

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

ADLER ALISON LEAL SOSTER

UM ESTUDO DE CASO SOBRE O DESENVOLVIMENTO DE AGENTES
INTELIGENTES PARA JOGOS DIGITAIS COM PARTICLE SWARM OPTIMIZATION

Joinville
2024

ADLER ALISON LEAL SOSTER

UM ESTUDO DE CASO SOBRE O DESENVOLVIMENTO DE AGENTES
INTELIGENTES PARA JOGOS DIGITAIS COM PARTICLE SWARM OPTIMIZATION

Trabalho apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica, no Curso de Engenharia Mecatrônica, do Centro Tecnológico de Joinville, da Universidade Federal de Santa Catarina.

Orientador: Prof. Dr. Pablo Andretta Jaskowiak

Joinville
2024

Dedico esse trabalho aos meus pais, Ádila R. L. Soster e Aldenir P. Soster, meus amigos, e todos os professores que permitiram-me aqui chegar.

AGRADECIMENTOS

Agradeço à Universidade Federal de Santa Catarina por prover acesso à educação superior de forma gratuita e com qualidade, permitindo-me o estudo da Engenharia Mecatrônica e formação profissional, agradeço pelas oportunidades inúmeras de crescimento profissional e pessoal.

Agradeço ao orientador deste trabalho, Prof. Dr. Pablo Andretta Jaskowiak, por revisar e contribuir por todo o processo de sua construção, sendo fundamental para sua conclusão.

Agradeço à minha família, em especial a meus pais, avôs e avós, pelo incondicional apoio durante toda a vida e formação, suas presenças fazem quem sou e a isso sou eternamente grato.

Aos amigos que me motivaram, agradeço a leveza e a alegria que permitiu a superação de momentos difíceis de minha carreira profissional e pessoal durante a jornada vivida.

RESUMO

Neste trabalho é realizado um estudo acerca da utilização de um algoritmo de otimização no treinamento de agentes para uma variante do jogo eletrônico Tetris. Na forma tradicional deste jogo, uma série de peças devem ser acomodadas em um tabuleiro 20 por 10, de forma a conseguir-se a maior pontuação possível, sendo possível a rotação e translação das peças para fazê-lo. Neste trabalho, é investigada a criação de um agente capaz de atingir o maior número de linhas completas em um jogo somente por meio da translação das peças, ou seja, nenhuma das peças pode ser rotacionada durante o jogo. Para obtenção desse agente, foi empregado o algoritmo de otimização conhecido por Particle Swarm Optimization (PSO).

Palavras-chave: Tetris. Agentes Inteligentes. Métricas.

ABSTRACT

This work is a study about the usage of an optimization algorithm in the training of intelligent agents for a variant of the Tetris video game. In its traditional implementation, a sequence of pieces must be accommodated in a 20 by 10 grid like board, so that one achieves the biggest possible score, being possible the translation and rotation of said pieces to do that. In this work, it's investigated the developing of an agent capable of achieving the highest number of completed lines in a game by only translating the pieces, that being that none of the pieces can be rotated during a game. To create this agent, an optimization algorithm is used, this algorithm is known as Particle Swarm Optimization (PSO)

Keywords: Tetris. Intelligent Agents. Metrics.

LISTA DE FIGURAS

Figura 1 – Captura de tela que mostra o jogo Tetris para a versão do NES	10
Figura 2 – Otimização para a Equação (6)	17
Figura 3 – Método utilizado para o PSO	22
Figura 4 – Tabuleiro com células de poço destacadas	23
Figura 5 – Tabuleiro com buracos marcados	23
Figura 6 – Tabuleiro com transições em colunas	24
Figura 7 – Tabuleiro com transições em linhas	25
Figura 8 – Tabuleiro com alturas de cada coluna	25
Figura 9 – Tabuleiro com demonstração do potencial de projeção	27
Figura 10 – Tabuleiro com demonstração das áreas contidas	27
Figura 11 – Tabuleiro com indicação de células que não contribuem para simetria	28
Figura 12 – Distribuição de linhas limpas para a avaliação média	34
Figura 13 – Otimização das métricas para o Subgrupo 4 para a avaliação média	34
Figura 14 – Distribuição de linhas limpas para a avaliação recorde	35
Figura 15 – Otimização das métricas para o Subgrupo 3 para a avaliação recorde	36
Figura 16 – Distribuição de linhas limpas para a avaliação ponderada	37
Figura 17 – Otimização das métricas para o Subgrupo 4 para a avaliação ponderada	37
Figura 18 – Otimização das métricas para o Subgrupo 1 para a avaliação média	44
Figura 19 – Otimização das métricas para o Subgrupo 2 para a avaliação média	45
Figura 20 – Otimização das métricas para o Subgrupo 3 para a avaliação média	45
Figura 21 – Otimização das métricas para o Subgrupo 4 para a avaliação média	46
Figura 22 – Otimização das métricas para o Subgrupo 5 para a avaliação média	46
Figura 23 – Otimização das métricas para o Subgrupo 1 para a avaliação recorde	47
Figura 24 – Otimização das métricas para o Subgrupo 2 para a avaliação recorde	47
Figura 25 – Otimização das métricas para o Subgrupo 3 para a avaliação recorde	48
Figura 26 – Otimização das métricas para o Subgrupo 4 para a avaliação recorde	48
Figura 27 – Otimização das métricas para o Subgrupo 5 para a avaliação recorde	49
Figura 28 – Otimização das métricas para o Subgrupo 1 para a avaliação ponderada	49
Figura 29 – Otimização das métricas para o Subgrupo 2 para a avaliação ponderada	50
Figura 30 – Otimização das métricas para o Subgrupo 3 para a avaliação ponderada	50
Figura 31 – Otimização das métricas para o Subgrupo 4 para a avaliação ponderada	51
Figura 32 – Otimização das métricas para o Subgrupo 5 para a avaliação ponderada	51

LISTA DE TABELAS

Tabela 1 – Configurações de execução do PSO para treino	30
Tabela 2 – Desempenho dos subgrupos para avaliação por média	33
Tabela 3 – Desempenho dos subgrupos para avaliação por recorde	35
Tabela 4 – Desempenho dos subgrupos para avaliação ponderada	36
Tabela 5 – Desempenho das diferentes avaliações do subgrupo 3	38
Tabela 6 – Desempenho das diferentes avaliações do subgrupo 4	38
Tabela 7 – Desempenho das diferentes avaliações do subgrupo 5	39
Tabela 8 – Desempenho das diferentes avaliações do subgrupo 1	52
Tabela 9 – Desempenho das diferentes avaliações do subgrupo 2	52
Tabela 10 – Desempenho das diferentes avaliações do subgrupo 3	52
Tabela 11 – Desempenho das diferentes avaliações do subgrupo 4	52
Tabela 12 – Desempenho das diferentes avaliações do subgrupo 5	53

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVOS	11
1.1.1	Objetivos Gerais	11
1.1.2	Objetivos Específicos	11
1.2	Organização do texto	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	Agentes	12
2.2	Algoritmos de otimização	13
2.2.1	Particle Swarm Optimization	14
2.3	Heurísticas	18
2.4	Agentes no contexto de jogos	19
3	MATERIAIS E MÉTODOS	21
3.1	Treinamento utilizando <i>PSO</i>	21
3.1.1	Utilização do modelo	21
3.1.2	Integração do modelo no jogo	22
4	DESENVOLVIMENTO	29
4.1	Como os experimentos foram realizados	29
4.1.1	Como os resultados são obtidos	29
4.2	OTIMIZAÇÃO DE MÉTRICAS	30
5	RESULTADOS E DISCUSSÃO	33
5.1	Avaliação dos Resultados	33
5.1.1	Avaliação dos subgrupos para a avaliação V_m	33
5.1.2	Avaliação dos subgrupos para a avaliação V_r	35
5.1.3	Avaliação dos subgrupos para a avaliação V_s	36
6	CONCLUSÕES	40
	REFERÊNCIAS	42
	APÊNDICE A	44
	APÊNDICE B	52

1 INTRODUÇÃO

Tetris é um jogo eletrônico criado em 1985 por Alexey Pajitnov (THIERY; SCHERRER, 2009), em que há um tabuleiro subdividido em células, as quais podem estar vazias ou preenchidas. O tabuleiro do jogo é organizado de forma que há 20 linhas de células e 10 colunas de células, formando um tabuleiro retangular com 200 células. Nesse jogo, sete tipos diferentes de peças que caem do topo, e essas devem ser posicionadas no tabuleiro. Quando uma linha do tabuleiro (10 células em uma mesma linha) está preenchida, a linha completa desaparece e rende pontos ao jogador, fazendo com que todas as células preenchidas acima da linha completada sejam movidas para baixo, dando continuação ao jogo, uma vez que as peças não possam surgir sem se sobreporem com células preenchidas, o jogo acaba. À esquerda da Figura 1, pode-se visualizar os diferentes tipos de peças do jogo, ao topo vê-se o número de linhas limpas durante aquele jogo, no caso, 21 linhas.

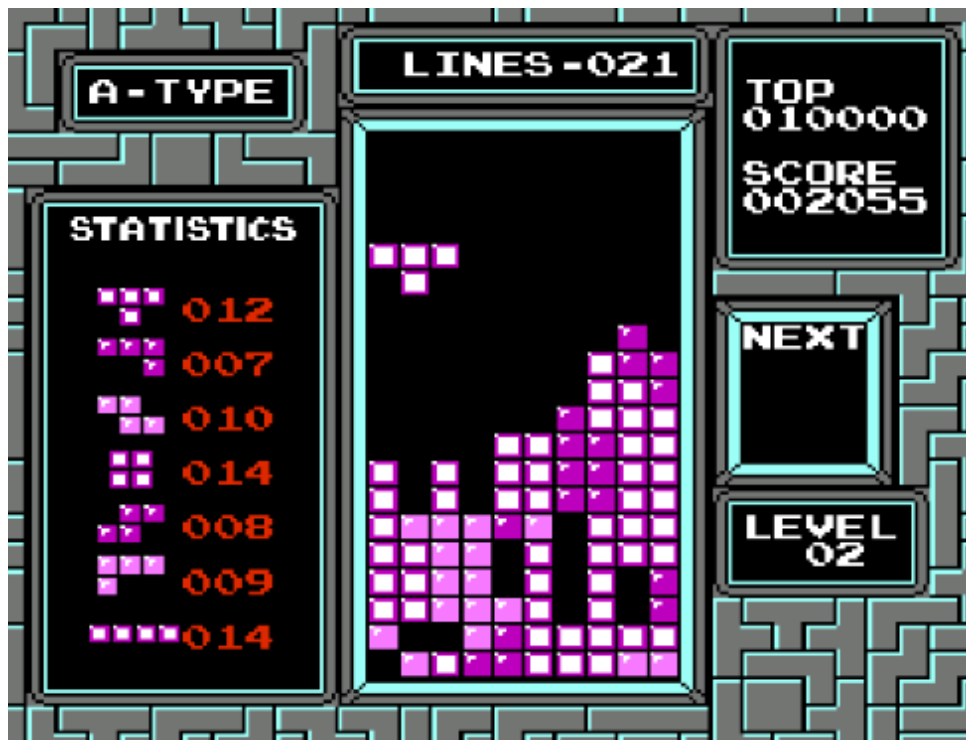
Conforme a pontuação do jogador aumenta, o nível aumenta, para elevar a dificuldade do jogo por meio da aceleração da queda das peças, com as quais o jogador deve lidar. A sequência em que essas peças, chamadas tetríminos, aparecem para o jogador é, para grande parte das diferentes versões do jogo, aleatória. Diversas implementações do jogo foram criadas desde sua concepção, sendo a versão utilizada para este trabalho desenvolvida em 1989 para o console Nintendo Entertainment System (NES), versão essa apresentada na Figura 1.

Com o sucesso do Tetris e suas diferentes versões, competições e variações surgiram durante as quase quatro décadas de existência do jogo. Uma das competições de maior relevância é o campeonato Classic Tetris World Championship (CTWC) (CTWC, 2023), que reuniu, por mais de dez anos, diversos competidores, jogando NES Tetris. Os jogadores participantes dessa disputa começam jogando no nível 18, e conseguem atingir o que é conhecido como *maxout*, uma pontuação tão alta, acima de 999,999 pontos, que a implementação do jogo se torna incapaz de registrar pontuações acima disso (SOSTER; BIRKEN; JASKOWIAK, 2021).

Graças ao sucesso atingido pelo Tetris, novas formas de jogá-lo foram criadas, uma das quais envolve o aumento da dificuldade através da adição de uma regra não originalmente atrelada ao jogo, onde nenhum dos tetríminos pode ser rotacionado durante o jogo. Essa regra, que depende dos jogadores não executarem o comando de rotação, dependeu esforços para a quebra dos recordes anteriormente atingidos, que no ano de 2021, era de 44 linhas removidas (SOSTER; BIRKEN; JASKOWIAK, 2021), o maior número, é atualmente de 72 linhas (BERECZKI, 2023).

Vale lembrar que com cada nível sucedido, a velocidade de queda dos tetríminos aumenta, tornando o jogo mais difícil de ser jogado pela velocidade na

Figura 1 – Captura de tela que mostra o jogo Tetris para a versão do NES



Fonte: Elaborado pelo autor (2024).

qual o jogador deve controlar a queda das peças e tomar a decisão de como fazê-lo. Por conta disso, os recordes humanos sem rotação são iniciados no nível 0, nível mais baixo e conseqüentemente lento e fácil do jogo. Esse fato torna-se irrelevante quando um sistema eletrônico, como um computador atual, com capacidade e velocidade suficiente joga, uma vez que a diferença entre o tempo de queda das peças não representa um problema na tomada de decisões de um sistema eletrônico.

Baseados nessa popularidade e competições, Soster, Birken e Jaskowiak (2021), realizaram uma série de testes com diversas variáveis para a produção de um agente capaz de jogar a variante sem rotações do tetris. Para isso, um algoritmo de otimização usado por esses foi o *Particle Swarm Optimization* (PSO), utilizado para encontrar a melhor solução para cada uma das heurísticas das quais utilizavam diferentes métricas, para desenvolver um agente capaz de atingir uma pontuação de 41 linhas. Como investigação, neste trabalho busca-se melhorar o desempenho do agente por meio da proposição de novas métricas mantendo o contexto sem rotações, com objetivo de tornar o espaço de busca melhor com a utilização de subconjuntos que podem exaltar o comportamento desejado do agente desenvolvido.

Esse estudo é desejado por conta do processo complexo de decisão para o posicionamento das peças, que deve ser guiado de alguma forma para o melhor desenvolvimento das ações do agente. O desenvolvimento de métricas demanda conhecimento do problema a ser resolvido, essas foram avaliadas ainda, para o

jogo com rotações. Dessa forma, busca-se desenvolver métricas que serão utilizadas especificamente para a variante sem rotações do tabuleiro e seus atributos.

1.1 OBJETIVOS

1.1.1 Objetivos Gerais

Avaliar novas métricas e conjuntos de métricas sob o contexto do jogo Tetris (Tetris Holding, 2023) sem rotações.

1.1.2 Objetivos Específicos

- a Propor novas métricas no contexto sem rotações para uso com o agente;
- b Implementar métricas propostas e avaliá-las utilizando o algoritmo de otimização Particle Swarm Optimization;
- c Avaliar os resultados atingidos.

1.2 ORGANIZAÇÃO DO TEXTO

Esse trabalho é organizado de forma que no Capítulo 2 são apresentados trabalhos essenciais para o embasamento deste trabalho. O Capítulo 3 elucida a metodologia utilizada, as métricas propostas e outras informações relevantes definidas. O Capítulo 4 apresenta informações acerca da obtenção dos resultados como as configurações utilizadas para os algoritmos. No Capítulo 5 apresenta-se os resultados atingidos por esse trabalho e suas conseqüentes discussões. Por fim, no Capítulo 6, encontram-se as conclusões provenientes dos resultados e discussões acerca de possíveis futuros trabalhos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados conceitos fundamentais para o entendimento do trabalho realizado, assim como trabalhos relacionados provenientes da literatura. Para prover entendimento dos métodos implementados são discutidos conceitos como agentes, heurísticas e algoritmos de otimização.

2.1 AGENTES

Para Russell e Norvig (2016), agentes são definidos por sua capacidade de obter informações sobre o ambiente ao qual estão sujeitos por meio de sensores, assim como agir sobre esse pela utilização de atuadores, definindo suas ações por escolhas informadas por dados que obtêm ou obtivera do espaço que estão inseridos, não utilizando informações da qual não sensorearam, ou seja, agentes não pode ser oniscientes.

Os autores definem algumas categorias que abrangem quase em totalidade os princípios de agentes inteligentes, os agentes reativos simples, agentes reativos baseados em modelo, agentes baseados em objetivos, e agentes baseados em utilidade. Agentes reativos simples são aqueles que realizam a decisão de atuação baseada no sensoriamento atual, sem considerar nenhum tipo de histórico. Agentes reativos baseados em modelo são aqueles que mantêm registros de informação baseados em sensoriamentos anteriores. Agentes baseados em objetivos são aqueles que para decidir necessitam ter um objetivo, como uma localidade à qual se deseja chegar, o que pode envolver busca de diferentes caminhos e planejamento, tornando-se mais flexíveis que agentes reativos pela possibilidade de alteração do objetivo, sendo o objetivo binário, satisfeito ou não. Agentes baseados em utilidade são mais flexíveis no sentido de atribuírem desejabilidades diferentes para formas diferentes de satisfazerem seus objetivos (RUSSELL; NORVIG, 2016).

Ainda, os agentes, por responderem aos estímulos que captam, devem ser capazes de realizar a ação desejada em resposta a essa. No exemplo de um agente que considera apenas o estado atual do ambiente, esse deve para cada sensoriamento, responder com uma ação apropriada, uma forma de fazê-lo se trata de construir uma tabela de consulta com pares sensoriamento-resposta, para definir a resposta mais apropriada. Esses agentes simples não são capazes, eficientemente, de resolver problemas complexos, como, por exemplo, jogar xadrez (RUSSELL; NORVIG, 2016).

Vale salientar que agentes são a união de duas partes, sua arquitetura e seu programa. A arquitetura de um agente se refere aos dispositivos físicos em que seu programa será executado, como o dispositivo de computação, sensores e atuadores. Por conta disso, problemas complexos não são passíveis de terem todas as possibilidades

mapeadas em uma tabela, o que torna essencial um método mais eficiente de decisão por parte do agente implementado, por conta da inexistência de dispositivos capazes de armazenar informações com tantos dados (RUSSELL; NORVIG, 2016).

Agentes para Nareyek (2001), possuem quase que mesma definição dos autores anteriores, uma vez que agentes são definidos por sua capacidade de sensorar e responder, de acordo com seus objetivos, no ambiente em que se inserem. O uso de agentes computacionais, ocorre uma vez que suas implementações permitem o estudo de problemas em um ambiente controlado e de fácil validação. Mesmo que caracterizados como um ambiente controlado, jogos oferecem uma série de características que podem ser encontradas em problemáticas reais, como tempo limitado para a tomada de decisões, ambiente dinâmico, recursos limitados e o conhecimento incompleto das variáveis que regem o ambiente. Agentes são classicamente utilizados em resoluções de problemáticas em jogos eletrônicos (NAREYEK, 2001). E para fazer o uso de agentes nesse contexto, alguns métodos computacionais são necessários, tais como algoritmos de otimização.

2.2 ALGORITMOS DE OTIMIZAÇÃO

Um agente capaz de aprendizado deve se tornar mais competente na tarefa para qual é treinado em fazer, assim, esse precisa de feedback utilitário para seu funcionamento, de forma que saiba a qualidade de seu desempenho. Problemas mais complexos, que dependem de uma série de passos para satisfação do objetivo, podem contar com algoritmos que auxiliam o agente a melhorar suas decisões para satisfazer o objetivo (RUSSELL; NORVIG, 2016).

Para o agente escolher entre diferentes possibilidades, feedback para suas ações deve ser obtido (RUSSELL; NORVIG, 2016), assim cada item participante do feedback deve ter uma importância em relação aos outros itens. Um carro autônomo, por exemplo, deve priorizar a segurança de seus passageiros a chegar ao destino mais rápido, logo, para definir o nível de influência de cada métrica, são necessários pesos que serão utilizados para implicar na importância relativa de cada item sendo monitorado durante o processo de decisão, e para isso, utiliza-se de um algoritmo de otimização. Algoritmos de otimização são um conjunto de técnicas algorítmicas que buscam otimizar um conjunto de parâmetros para atingir uma solução ótima ou satisfatória para uma problemática, como, por exemplo, encontrar que em um ambiente de busca, o conjunto de pesos mais adequado para o que se deseja atingir (BANGERT, 2021).

Dos diferentes algoritmos disponíveis para a tarefa, o escolhido neste trabalho é baseado no comportamento social coletivo de enxames na natureza. Esses algoritmos são categorizados como algoritmos de inteligência de enxame por Gad (2022). Tais

algoritmos de enxame são, como dito anteriormente, inspirados no comportamento social de animais, como os voos de aves e o nado de cardumes, em que os movimentos de um indivíduo influência não apenas os movimentos de seus vizinhos como suas descobertas também podem desencadear o benefício de todos do grupo (KENNEDY; EBERHART, 1995). De acordo com Gad (2022), esses algoritmos são bem estabelecidos na literatura como escolhas sólidas para otimizar os mais diversificados processos nas mais variadas áreas industriais e do conhecimento. Desse grupo de técnicas, um dos mais populares e conhecidos é o de otimização por enxame de partículas ou *Particle Swarm Optimization*.

2.2.1 Particle Swarm Optimization

Esse algoritmo de otimização foi proposto por Kennedy e Eberhart (1995), para a otimização de funções não lineares por meio de metodologias de enxame. Esse algoritmo é fundamentado no comportamento de revoadas de pássaros em busca de milho, em que por meio da partilha de informações entre os pássaros, reúnem-se no ponto desejado. Para os pássaros o ponto desejado representa o alimento, para um algoritmo, a solução encontrada de melhor desempenho (POLI; KENNEDY; BLACKWELL, 2007). Os diferentes integrantes da rede social dividem informações, essa hipótese é fundamental para o desenvolvimento do método (KENNEDY; EBERHART, 1995), um sistema usado na otimização de soluções. No entanto, há dois elementos em que o algoritmo difere de sua inspiração do comportamento dos pássaros, as entidades envolvidas não colidem entre si e também não buscam manter velocidades de movimento semelhantes, categorizando-as não mais como uma revoada, mas sim como um enxame (KENNEDY; EBERHART, 1995).

Sua definição como enxame implica na satisfação de cinco princípios (WANG; TAN; LIU, 2018):

1. População deve ter o poder de realizar cálculos simples de espaço e tempo;
2. O enxame deve conseguir responder ao ambiente;
3. O enxame não deve limitar-se na busca de soluções, deve ser capaz encontrá-las de diferentes formas;
4. O enxame não deve alterar comportamento para toda mudança ambiental;
5. O enxame deve alterar de comportamento quando essa alteração vale a pena.

Assim, para o agente desenvolvido tomar decisões apropriadas, o algoritmo de otimização irá ser responsável por encontrar no espaço de busca, os pesos que estimulam o comportamento desejado. Esses pesos então são avaliados segundo a função de *feedback*, de avaliação desenvolvida para a limpeza de linhas, essa função é aquela que define que pesos são os mais apropriados para o comportamento desejado, essencial para o aprendizado.

O funcionamento do algoritmo é dado de forma que uma série de indivíduos extremamente simples, chamados partículas são dispostos no espaço de busca de forma aleatória, a cada iteração, as partículas se movem em busca do ponto ótimo da função, espaço de busca, em que estão submetidos (POLI; KENNEDY; BLACKWELL, 2007).

Para que essa movimentação ocorra, diversas informações são registradas para cada uma das n partículas dispostas no espaço de busca, como os vetores posição, velocidade e melhores soluções pessoais e global, s_n , v_n , s^p e s^g . O vetor s_n armazena a posição atualmente ocupada para uma dada partícula do espaço, sendo tal, uma das diversas possíveis soluções para o problema. O vetor v_n representa o vetor velocidade, que para cada iteração irá implicar na movimentação da partícula para uma nova posição. Por fim, o vetor s^p registra a melhor posição, solução para o problema, registrado por uma determinada partícula, enquanto s^g registra a solução mais adequada dentre todas as partículas (POLI; KENNEDY; BLACKWELL, 2007).

A cada iteração do algoritmo, a posição de cada uma das n partículas é avaliada, assim como é feita a verificação da existência de novas soluções mais apropriadas, que levam a uma atualização das equações de velocidade da partícula conforme a distância até a melhor solução da partícula e a melhor solução global conhecidas e outros coeficientes, como mostrado na Equação (1) (POLI; KENNEDY; BLACKWELL, 2007) (KENNEDY; EBERHART, 1995).

$$v_{i+1} = w_i v_i + c_i^1 r_i^1 (s_i^p - s_i) + c_i^2 r_i^2 (s_i^g - s_i) \quad (1)$$

Os termos da Equação (1) trabalham em conjunto para definir, a cada iteração, a velocidade da próxima iteração v_{i+1} , de forma com que três diferentes vetores são somados para obter essa informação. O primeiro vetor $w_i v_i$, formado pelo coeficiente w_i e vetor v_i , retrata a velocidade inercial da partícula, sua tendência a manter o movimento. O vetor $c_i^1 r_i^1 (s_i^p - s_i)$ representa a componente que direciona a velocidade da partícula para a melhor solução avaliada no espaço de busca, por essa mesma. Por fim a componente $c_i^2 r_i^2 (s_i^g - s_i)$ é a componente do vetor velocidade que atrai a partícula para o ponto de melhor avaliação dentre todas as soluções avaliadas pelas partículas no espaço de busca.

Para cada iteração i , a velocidade da partícula é atualizada conforme os vetores apresentados, cada um desses vetores é ponderado por um coeficiente escalar, sendo eles, w , coeficientes de inércia, c^1 , coeficiente cognitivo, e c^2 , o coeficiente social.

O coeficiente de inércia é visto como aquele com maior influência sobre o desempenho do algoritmo do PSO, esse influencia a tendência da partícula em manter seu movimento, e assim faz com que a partícula explore mais a função de maneira global, uma vez que com o aumento do coeficiente (WANG; TAN; LIU, 2018), a influência da velocidade atual da partícula aumenta para o cálculo da velocidade para a próxima

iteração.

O coeficiente cognitivo, também conhecido por fator de aceleração cognitivo, afeta o comportamento da partícula, para atrair o movimento da partícula em sentido ao ponto de melhor valor pessoal. O aumento desse coeficiente implica a uma relação inversamente proporcional à exploração do espaço pela partícula (WANG; TAN; LIU, 2018), isso ocorre uma vez que com o aumento do coeficiente, a velocidade da partícula é mais influenciada por esse termo da equação, que faz com que ela se torne mais “focada” nos valores que avaliou, em sua melhor solução local.

O coeficiente social, também conhecido por fator de aceleração social, afeta o comportamento da partícula, para atrair o movimento da partícula em sentido ao ponto de melhor valor global, dentre a melhor solução de todas as partículas. O aumento desse coeficiente implica a uma relação proporcional à exploração do espaço pela partícula (WANG; TAN; LIU, 2018), pelo aumento da influência do termo calculado com base na distância para a melhor solução global, que atrai a partícula para aquele ponto.

Os termos s^p e s^g representam, respectivamente, as melhores soluções encontradas para uma partícula e a melhor solução encontrada dentre todas as partículas utilizadas na busca. A melhor resposta de uma partícula é aquela que representa a melhor solução avaliada para aquela partícula. A melhor resolução global é obtida por meio da comunicação com todas as partículas envolvidas, em que a melhor solução encontrada dentre todas as soluções encontradas por cada uma das partículas é armazenada.

Além disso, os termos r_i^1 e r_i^2 são números randômicos gerados a cada iteração, contidos no intervalo [0; 1]. Assim como esses coeficientes randômicos se alteram, os coeficientes c_i^1 e c_i^2 e w_i são atualizados de acordo com as Equações (2), (3), e (4), em que o valor desses coeficientes é gradualmente alterado durante o processo de otimização (POLI; KENNEDY; BLACKWELL, 2007), em que os coeficientes social e cognitivo tem seu valor aumentado até seu valor máximo durante iterações, enquanto o coeficiente de inércia diminui até seu valor mínimo.

$$w_i = w_{min} + (w_{max} - w_{min}) \frac{i_{max} - i}{i_{max}} \quad (2)$$

$$c_i^1 = (c_{final}^1 - c_{inicial}^1) \frac{i}{i_{max}} + c_{inicial}^1 \quad (3)$$

$$c_i^2 = (c_{final}^2 - c_{inicial}^2) \frac{i}{i_{max}} + c_{inicial}^2 \quad (4)$$

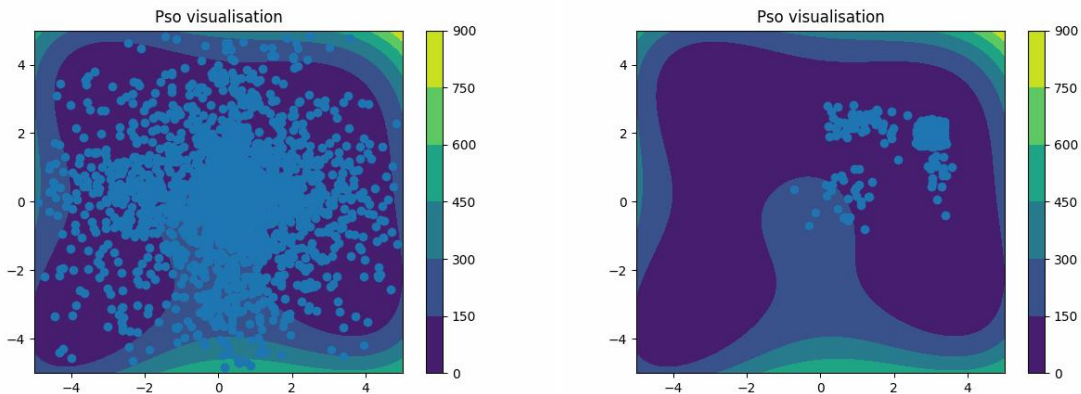
A Equação (5) apresenta o método de cálculo para a posição para uma dada iteração, com base na cinemática, onde a nova posição é dada pela posição atual somada a velocidade do agente.

$$s_{i+1} = s_i + v_i \quad (5)$$

Para exemplificar o funcionamento do algoritmo de otimização, o autor deste trabalho define códigos em Python para demonstrar como as partículas se comportam submetidas no espaço de busca. Para isso, utiliza-se a função definida pela Equação (6), para buscar-se o ponto de mínimo. Dessa forma, o comportamento do sistema pode ser exemplificado pela Figura 2. Na primeira parte dessa, à esquerda, vê-se a posição inicial das partículas, primeiramente definida de forma aleatória, com a passagem de diversas iterações durante o processo de otimização, as partículas vão a convergir para um dos quatro mínimos da função objetivo, para encontrar uma solução não ótima, porém extremamente próxima ao valor mínimo da função, zero, o que é demonstrado na segunda parte da figura, à direita da imagem.

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \quad (6)$$

Figura 2 – Otimização para a Equação (6)



Fonte: Elaborado pelo autor (2024).

A função usada como exemplo na Figura 2 foi otimizada de modo a encontrar como valor mínimo após a convergência das partículas, o valor de $3.1867 \cdot 10^{-19}$, considerado matematicamente não ótimo, uma vez que o valor ótimo é 0, mas extremamente satisfatório para a demonstração por mostrar a característica de encontro de uma solução, potencialmente não ótima, porém satisfatória.

Para demonstrar a simplicidade do algoritmo PSO claramente, a implementação desenvolvida para essa demonstração é apresentada na forma de pseudocódigo com objetivo de abstrair especificidades de linguagens de programação e permitir entendimento por parte do leitor.

```
1 INICIALIZAR parametros do PSO:
2 (N, c1, c2, Wmin, Wmax, Vmax, maxIter)
3
4 DEFINIR limites do espaco de busca
5 INICIALIZAR todas as N particulas
6
7 PARA cada particula
8     CALCULE o objetivo
9     ATUALIZE o recorde pessoal se melhor
10    ATUALIZE o recorde global se melhor
11 FIM
12
13 ATUALIZE coeficiente de inercia
14     PARA cada particula
15         ATUALIZE a velocidade
16         Atualize a posicao
17     FIM
18 ENQUANTO a condicao nao for satisfeita
19 RETORNE a melhor solucao
```

2.3 HEURÍSTICAS

Heurísticas são artifícios utilizados na computação, para permitir a realização de escolhas não completamente ótimas para se alcançar uma decisão para solucionamento de um problema desejado, para guiar o desenvolvimento da solução. Essas, são utilizadas em lugar de soluções exatas, principalmente em problemas de otimização difíceis, para a elaboração de estratégias em tempo computacional e resultados eficientes (OLIVEIRA et al., 2016).

Para a utilização de um agente, um dos possíveis métodos a serem utilizados é o emprego de variáveis a serem avaliadas para tomadas de decisão, tais variáveis são chamadas de métricas. As heurísticas utilizadas pelo agente definem uma função objetivo que deve ser maximizada ou minimizada, neste trabalho, serão maximizadas.

As heurísticas podem ser categorizadas como heurísticas construtivas e de melhoria. Construtivas são aquelas utilizadas para desenvolver soluções adicionando elementos, item a item, em que caso o processo seja interrompido não será entregue solução alguma (OLIVEIRA et al., 2016). Para o trabalho desenvolvido, serão utilizadas métricas de melhoria, que serão discutidas em futuras seções.

Heurísticas de melhoria, no entanto, são iniciadas com uma solução obtida de

forma aleatória ou com construtivas, aplicando alterações na solução, para atingir-se um critério definido anteriormente (OLIVEIRA et al., 2016).

A utilização de métricas se faz importante especialmente para problemáticas sem um comportamento determinístico ou uma solução ótima conhecida, isso é exposto por (DEMAINE; HOHENBERGER; LIBEN-NOWELL, 2003), o qual comprova que Tetris é um jogo com características *NP-hard* e *NP-Complete*, de difícil obtenção da solução.

No contexto do Tetris, agentes de uma peça são agentes que consideram apenas o tetrimino atual em suas decisões, agentes de duas peças consideram a peça atual e a próxima, mostrada ao lado do tabuleiro para o jogador. Um agente conhecido de uma peça, é o agente de Pierre Dellacherie (2003), considerado por um tempo considerável como o melhor agente de uma peça do mundo por parte da comunidade envolvida, esse utiliza um conjunto de métricas que descrevem o estado do tabuleiro, em que a importância de cada uma dessas métricas foi manualmente ajustada pelo seu autor para obter o comportamento desejado (FAHEY, 2003).

2.4 AGENTES NO CONTEXTO DE JOGOS

Como dito anteriormente, agentes são classicamente usados em jogos (NAREYEK, 2001), um exemplo de agente no contexto de jogos é o agente desenvolvido por OpenAI et al. (2019) para jogar Dota 2 (Valve Corporation, 2024), jogo de grande complexidade com dez jogadores separados em dois times rivais, em que o objetivo é destruir a base do outro time. Esse agente criado por eles derrotou o time campeão mundial, superando o desempenho humano para esse jogo.

No contexto do jogo Tetris, um dos agentes de nota é aquele desenvolvido por Pierre Dellacherie, que desenvolveu um agente capaz de jogar Tetris com a utilização de 6 simples métricas que tiveram seus pesos ajustados manualmente para um posterior cálculo linear. Esse agente foi considerado o melhor agente jogador de tetris não humano até o ano de 2008, por meio da utilização de uma heurística de tomada de decisões utilizando métricas e pesos para a tomada de decisões (ALGORTA; ŞİMŞEK, 2019).

Outro agente notável no contexto do Tetris é aquele desenvolvido por Böhm, Kóokai e Mandl (2005) que utilizou métricas comuns em outros trabalhos, ainda realizou o uso de novas métricas, com diferentes métodos de avaliação. Os autores seguiram uma heurística de avaliação dos possíveis tabuleiros para as peças e assim, com a função de avaliação, a qual utiliza as métricas ponderadas por pesos para a tomada de decisão de qual é o melhor tabuleiro, e assim atingi-lo. O agente desses diferentemente, do criado por Dellacherie, utiliza como algoritmo de otimização, um algoritmo genético, intimamente relacionado com PSO (WANG; TAN; LIU, 2018).

Birken (2019) por sua vez, programou um agente capaz de jogar em

competência semelhante e talvez superior ao de um jogador humano, uma vez que o agente desenvolvido é capaz de atingir *max out* com probabilidade de 80% dos jogos jogados. Para otimizar as decisões do agente, o autor utilizou de PSO para definir os pesos das 17 métricas utilizadas. O agente desenvolvido pôde alcançar pontuação de $1\,036\,706 \pm 149\,458$.

O programa desenvolvido por Birken (2019) seguiu para ser modificado e utilizado por Soster, Birken e Jaskowiak (2021) para uma variante do jogo original, em um contexto sem rotações, em que três diferentes subgrupos foram avaliados sobre as mesmas configurações de treinamento e avaliação. Desses grupos, um conjunto de métricas formado por 7 métricas provenientes da união do conjunto de métricas, totalizando 17 diferentes métricas, propostas por 3 autores que definiram uma abordagem heurística baseada em agentes computacionais para jogar Tetris com rotações.

3 MATERIAIS E MÉTODOS

Neste capítulo há os conteúdos acerca dos métodos utilizados na produção do trabalho, para a construção dos códigos, decisões, ferramentas utilizadas e desenvolvimento para obtenção dos resultados. Além disso, são explicadas informações como os parâmetros definidos para o funcionamento do projeto, explicações dos métodos utilizados e suas justificativas.

3.1 TREINAMENTO UTILIZANDO *PSO*

A otimização do uso das métricas definidas ocorre pela definição de diferentes subconjuntos de métricas, assim como a proposição de novas métricas a serem utilizadas em aspiração de obter subconjuntos mais informativos para o comportamento do agente para a problemática especificada. Para realizar os ajustes necessários, o algoritmo do *PSO* necessita que uma série de parâmetros sejam definidos, os quais irão definir o comportamento, assim como restringir os limites do algoritmo.

3.1.1 Utilização do modelo

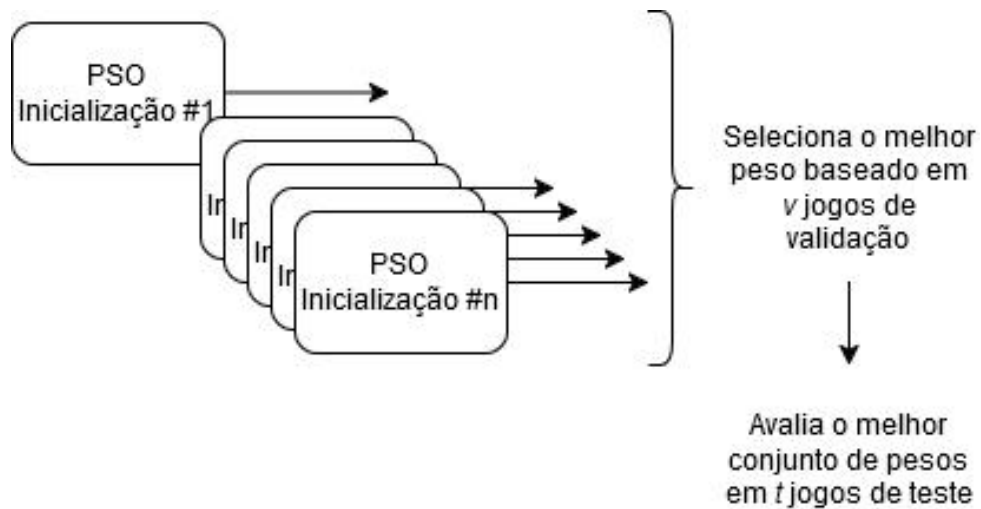
O comportamento do agente a ser estimulado é o de maximizar o número de linhas completas em um único jogo, assim sendo, deve-se otimizar os pesos no espaço de busca para as métricas extraídas do tabuleiro, para que esse comportamento seja priorizado.

A execução do *PSO* é realizada de forma que, para cada partícula, seu desempenho é avaliado em 25 jogos, de forma que as decisões tomadas utilizando a solução da partícula tem sua posição avaliada por meio da sua função de avaliação. Assim, a posição que maximiza a avaliação é escolhida, ambas as peças em queda no tabuleiro quanto a próxima, conhecida, são utilizadas para a tomada de decisão. Para cada partícula inserida no sistema, os pesos utilizados na função de avaliação podem ser diferentes, implicando em distintos comportamentos para as partidas jogadas. As partículas que mais se destacam na série de jogos de treinamento, são consideradas soluções mais apropriadas. Três diferentes métodos de validação são considerados dentre os jogos de validação é utilizada para o agente ser avaliado, para considerar também como os pesos serão otimizados.

O processo de otimização do *PSO* é repetido diversas vezes, de forma que, devido à natureza estocástica do método, seja possível obter uma solução com um melhor desempenho. A Figura 3 ilustra o procedimento de treinamento.

Para permitir uma avaliação precisa e comparável entre as diferentes soluções encontradas, os jogos de validação são os mesmos entre as diversas validações. O

Figura 3 – Método utilizado para o PSO



Fonte: Soster et al. (2021)

mesmo ocorre para os jogos de teste, diferentes dos jogos de validação, e ainda iguais entre os testes realizados.

3.1.2 Integração do modelo no jogo

O algoritmo de otimização é integrado por meio do desenvolvimento de um novo pacote para o projeto, o qual visa utilizar ao máximo o código desenvolvido por (BIRKEN, 2019)

Baseado nos trabalhos anteriormente desenvolvidos, as métricas enumeradas de 1, 2, 3, 4, 5, 6 e 7, são propostas em subgrupos já testados (SOSTER; BIRKEN; JASKOWIAK, 2021), enquanto durante o processo são propostas as outras métricas enumeradas, sendo cada uma delas listada e logo após explicadas para o bom entendimento.

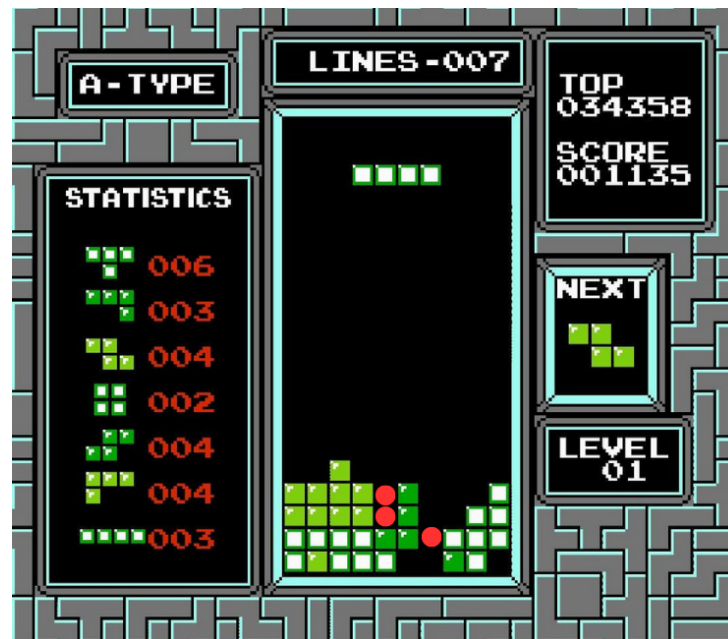
1. Altura total de fixagem:

Soma das alturas em que ambos os tetriminos conhecidos foram fixados;

2. Número total de células poço:

Somatório de todas as células de poços. Um poço é uma célula vazia cercada de células preenchidas em exceção ao seu topo, as paredes do tabuleiro são consideradas preenchidas. A Figura 4 apresenta as células que contribuem para essa métrica destacadas por círculos vermelhos;

Figura 4 – Tabuleiro com células de poço destacadas

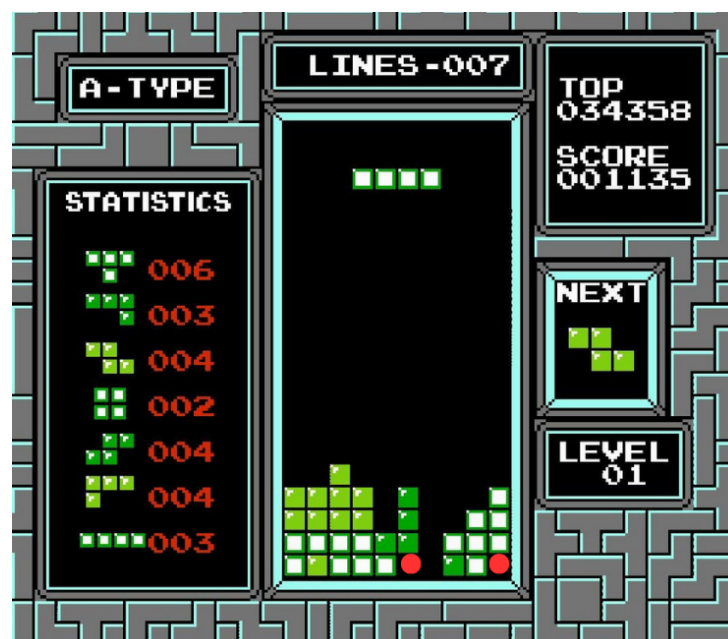


Fonte: Elaborado pelo autor (2024)

3. Número de buracos em colunas:

Número de células vazias com células preenchidas imediatamente acima, colunas vazias não possuem buracos. A Figura 5 apresenta os buracos do tabuleiro marcados em vermelho;

Figura 5 – Tabuleiro com buracos marcados

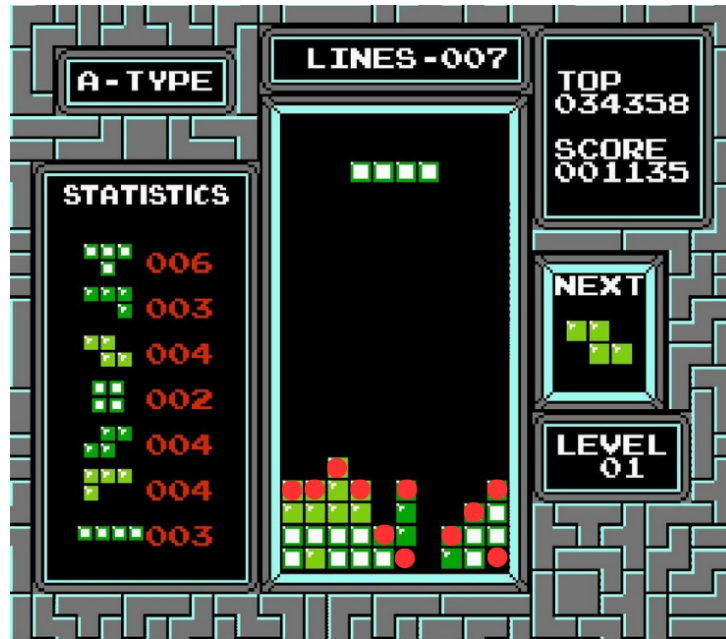


Fonte: Elaborado pelo autor (2024)

4. Número de transições de colunas:

Soma de transições entre célula preenchida e vazia, e vice-versa para colunas. Transições nas bordas, topo e chão não contabilizam. A Figura 6 apresenta um exemplo;

Figura 6 – Tabuleiro com transições em colunas

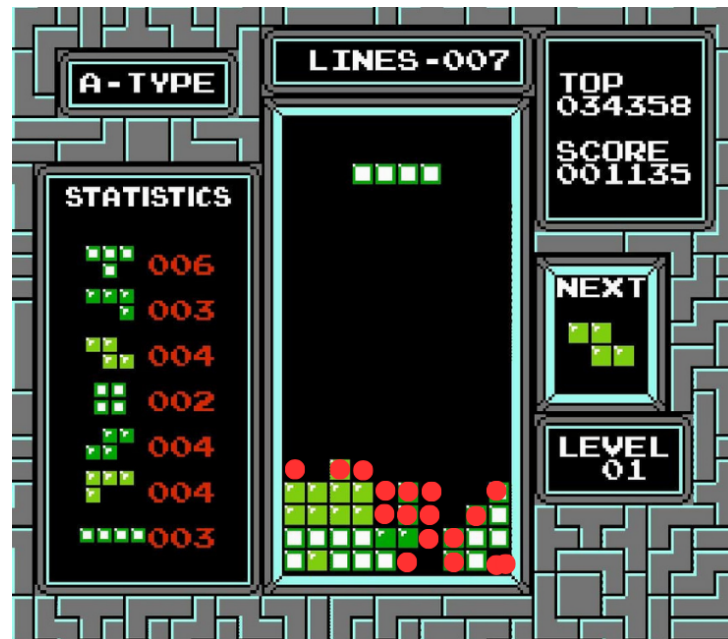


Fonte: Elaborado pelo autor (2024)

5. Número de transições de linhas:

Soma de transições entre células preenchidas e vazias, e vice-versa, para linhas. As paredes do tabuleiro são contabilizadas como células preenchidas, linhas vazias são ignoradas. A Figura 7 apresenta uma visualização dos locais na qual a métrica seria incrementada para aquele tabuleiro;

Figura 7 – Tabuleiro com transições em linhas

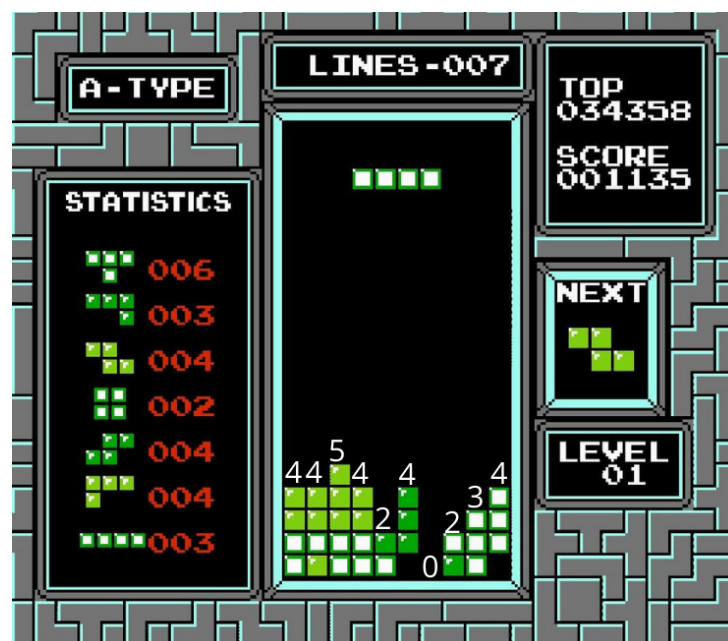


Fonte: Elaborado pelo autor (2024)

6. Somatório de alturas:

O somatório das diferentes alturas de cada coluna, altura sendo a maior distância entre uma célula preenchida e o chão do tabuleiro para cada coluna, sendo que colunas vazias possuem uma altura de zero. Para o tabuleiro da Figura 8, o resultado é 32;

Figura 8 – Tabuleiro com alturas de cada coluna



Fonte: Elaborado pelo autor (2024)

7. Intervalo de alturas:

A diferença de altura, entre a coluna mais alta e mais baixa, para o tabuleiro da Figura 8, o resultado é 5;

8. Centro de massa vertical:

Somatório ponderado das células preenchidas do tabuleiro, em que os pesos são dados pela altura da célula somada;

9. Centro de massa horizontal:

Somatório ponderado das células preenchidas do tabuleiro, em que os pesos são dados pela distância à coluna central, colunas à esquerda contribuem negativamente, e à direita positivamente;

10. Porcentagem de preenchimento:

Razão entre células preenchidas e número total de células, representando a porcentagem preenchida no tabuleiro em um intervalo de 0 a 1;

11. Entropia do tabuleiro:

Calculado pela entropia do tabuleiro, a fórmula utilizada para a entropia é apresentada na Equação (7), essa equação é proveniente do campo da teoria de informação (ROCHOL, 2009), em que ocorre o somatório das probabilidades p_i de uma dada célula i estar preenchida ou vazia multiplicada pelo logaritmo de base 2 dessa probabilidade;

$$H(x) = - \sum_{i=1}^N p_i \cdot \log_2(p_i) \quad (7)$$

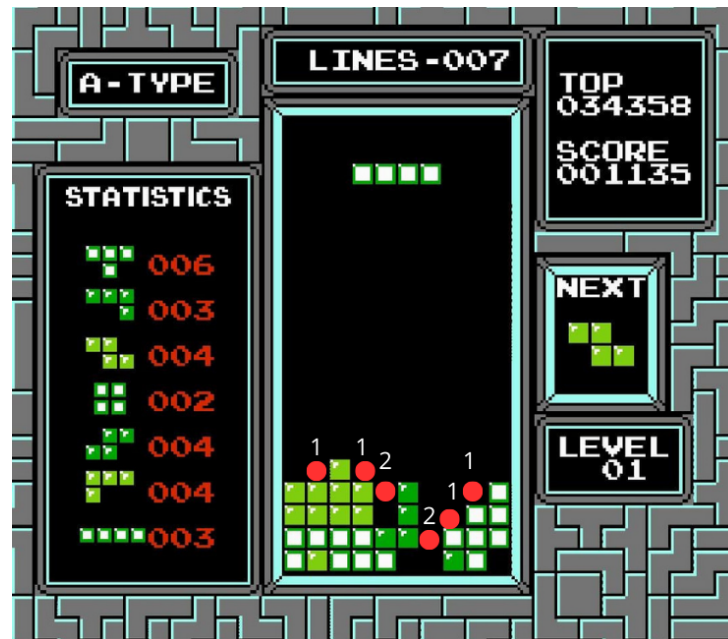
12. Chance de trava em cantos:

A chance de trava em cantos define uma pontuação para células trancadas nos cantos do tabuleiro por uma célula preenchida, ou seja, na coluna mais a esquerda e mais à direita do tabuleiro;

13. Potencial de projeção:

Essa métrica é calculada através da comparação de cada coluna com suas colunas adjacentes, caso a coluna tenha uma altura menor que as adjacentes, o valor da métrica é incrementado. A Figura 9 apresenta como a configuração do tabuleiro influenciaria essa métrica, em que os números da imagem acima dos círculos vermelhos representam a contribuição da coluna para a métrica.

Figura 9 – Tabuleiro com demonstração do potencial de projeção

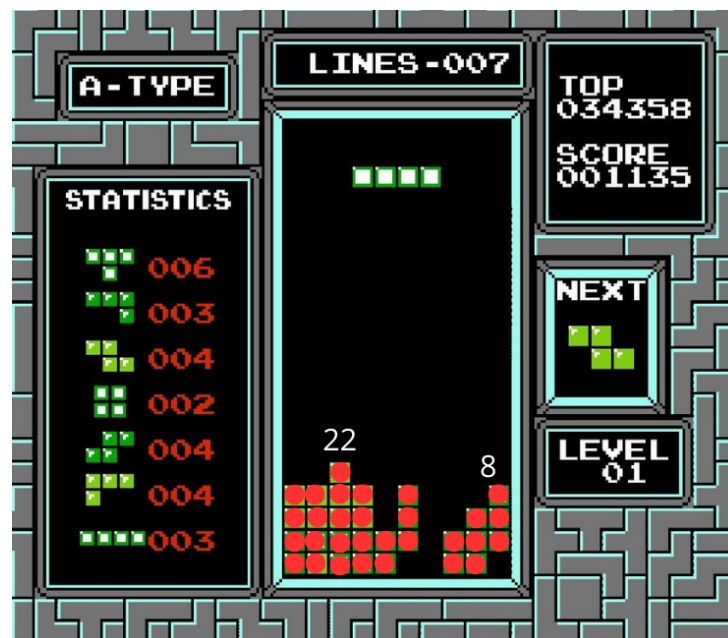


Fonte: Elaborado pelo autor (2024)

14. **Área máxima contínua:**

Armazena o maior valor de área preenchida continuamente no tabuleiro. Para o tabuleiro da Figura 10, o resultado é 22;

Figura 10 – Tabuleiro com demonstração das áreas contidas



Fonte: Elaborado pelo autor (2024)

15. **Área mínima contínua:**

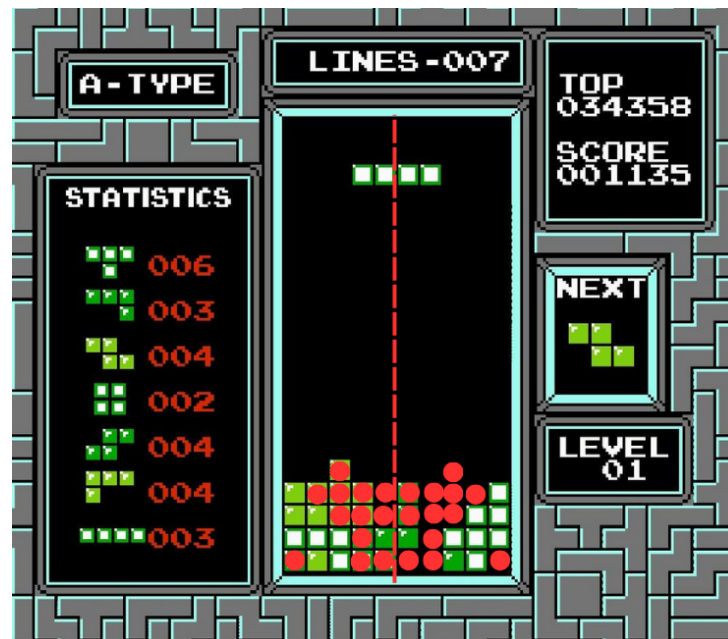
Armazena o menor valor de área preenchida continuamente no tabuleiro. Para o

tabuleiro da Figura 10, o resultado é 8;

16. Simetria do tabuleiro:

Avalia o quão simétrico é o tabuleiro, considerando a coluna do meio como o plano de espelhamento. A Figura 11 apresenta um tabuleiro na qual as células que não contribuem para a simetria estão destacadas, o eixo de simetria é apresentado pelo seguimento de linha vertical.

Figura 11 – Tabuleiro com indicação de células que não contribuem para simetria



Fonte: Elaborado pelo autor (2024)

17. Número de possíveis posições:

Essa métrica avalia quantas posições os diferentes tetriminos são capazes de ocupar sem colidir com células já preenchidas

4 DESENVOLVIMENTO

Para realizar os experimentos, utilizam-se ferramentas de software gratuitas e hardware já disponíveis para o autor desse trabalho, essas restrições são fundamentais em uma série de decisões que impactam a complexidade computacional do processo de otimização e as soluções adotadas. Essas limitações demandaram ajustes nas abordagens utilizadas, para viabilizar a produção do trabalho.

4.1 COMO OS EXPERIMENTOS FORAM REALIZADOS

Ao final de cada avaliação dos subconjuntos de métricas, seus resultados são armazenados para posterior análise. Esses resultados incluem informações sobre os pesos relativos entre cada uma das métricas do subconjunto, bem como o número de linhas limpas pela melhor solução encontrada. Para a análise, os diferentes subconjuntos são sistematicamente comparados para explicitar os resultados obtidos. A sua comparação permite verificar quais métricas contribuíram de forma positiva com maior significância para o comportamento do agente, exaltando quais podem ser utilizadas com maior eficácia em futuras implementações.

Como mencionado anteriormente, o hardware utilizado na produção do trabalho pertence ao autor. São apresentados em seguida as especificações do mesmo, para fins de clarificação e posteridade.

Para realizar a execução dos softwares desenvolvidos, faz-se utilizar um computador Windows, com um processador, Intel(R) Core(TM) i5-9300H 2.40GHz, 16 GB de RAM e uma NVIDIA GeForce GTX 1050 3 GB. Esse hardware foi usado para o treinamento, avaliação e teste dos subgrupos, executando os testes por diversas horas. Para realizar os experimentos, foi definido o projeto para seguir a estrutura da ferramenta *Maven* (The Apache Software Foundation, 2024), o software *Netbeans* é utilizado para executar o processo. Para a produção de gráficos, são implementados códigos utilizando *Python* (Python Software Foundation, 2024) e bibliotecas como *Matplotlib* (The Matplotlib development team, 2024) para melhor visualização e comparação dos resultados.

4.1.1 Como os resultados são obtidos

Para desenvolver o trabalho, é utilizada uma base de códigos desenvolvida por Birken (2019), que implementa o jogo, com as regras padrões. Logo, para adicionar as restrições desejadas, ausência de rotações, esse código foi modificado para forma a não permitir a rotação dos tetriminos.

A implementação desenvolvida permite que se jogue o jogo sem o uso de interfaces gráficas, permitindo maior velocidade na computação dos jogos.

4.2 OTIMIZAÇÃO DE MÉTRICAS

Para simplificar o processo de avaliação, os diferentes grupos serão avaliados com hiper-parâmetros reduzidos para a comparação final, dessa forma, a execução da otimização é agilizada, permitindo maior velocidade na comparação dos diferentes grupos.

Assim pode-se comparar os hiper-parâmetros de treinamento e teste final através da visualização promovida pela Tabela 1.

Os parâmetros necessários para definir o comportamento do algoritmo do PSO podem ser visualizados na Tabela 1, a qual apresenta esses hiper parâmetros e seus valores definidos com uma breve descrição de cada para o completo e inequívoco entendimento desses.

Tabela 1 – Configurações de execução do PSO para treino

Parâmetro	Valor
Número máximo de iterações	150
Número de partículas	30
Intervalo do coeficiente de inércia	0,9 / 0,4
Intervalo do coeficiente cognitivo	1,5 / 0,5
Intervalo do coeficiente social	1,0 / 4,0
Intervalo de busca dos pesos	[-1; 1]
Intervalo de velocidade da partícula	[-0,01; 0,01]
Número de jogos de treinamento	25
Número de jogos de validação	250
Número de jogos de teste	20 000
Número de reinicializações	5

Fonte: Elaborado pelo autor (2024)

O *número máximo de iterações* define quantas iterações poderão ocorrer para uma dada inicialização do algoritmo de PSO. *Intervalo do coeficiente de inércia* definem o intervalo, os valores máximos e mínimos, que o coeficiente pode atingir durante o processo de decaimento, de um valor inicial máximo para mínimo ao longo das iterações, mínimos que o coeficiente pode atingir durante o decaimento que ocorre no processo. De mesma forma, o *intervalo do coeficiente cognitivo*, define os valores de decaimento, inicialmente máximo, para o mínimo para o coeficiente. O *intervalo do coeficiente social* define seus valores de progressão inicial e final, assim como ocorre para os outros coeficientes. O *intervalo de velocidade* da partícula define os limites de velocidade positiva e negativa para cada uma das partículas no espaço de busca. O

parâmetro *Número de jogos de treinamento* define o número de jogos nos quais cada uma das partículas será avaliada durante o treinamento para cada iteração de cada uma das inicializações. O *número de jogos de validação* define o número de jogos a serem utilizados durante a validação do treinamento de uma dada inicialização, para os jogos definidos, somente uma, a melhor solução da inicialização sendo validada é executada. Por fim, o *número de jogos de teste* define o número de jogos que serão utilizados para avaliar a melhor solução encontrada dentre todas as inicializações do algoritmo.

Para cada grupo subconjunto de métricas avaliado, o número de partículas é definido como 30. A escolha em manter o número de partículas com esse valor, é para manter uma objetividade de comparação entre os diferentes grupos.

Ainda, para cada subgrupo, três diferentes avaliações serão realizadas, nessas, diferentes versões da função de avaliação são usadas, uma delas considera ao recorde entre todos os jogos de treinamento (V_r), uma considera a média de linhas (V_m), uma considera um valor baseado pela média ponderada entre o recorde de linhas limpas e a média de linhas limpas (V_s), no qual que o número de linhas limpas para cada jogo é contido em L e os pesos para o recorde é dado por p_r e o peso para a média p_m entre todos os jogos de treinamento, a equação que fornece esse valor é vista na Equação (8). Os valores de p_r e p_m são definidos por 0.4 e 0.6 respectivamente, para incentivar um desempenho mais uniforme dentre diferentes jogos.

$$\text{Pontos} = p_r \cdot \max(L) + p_m \cdot \text{mean}(L) \quad (8)$$

O primeiro subgrupo otimizado considera as métricas avaliadas em trabalhos anteriores (SOSTER; BIRKEN; JASKOWIAK, 2021), as métricas que compõem este grupo são aquelas, de números, 1, 2, 3, 4, 5, 6, 7. Ou seja, *Altura Total de Fixagem* (ATF), *Número Total de Células Poço* (NTCP), *Número de Buracos em Colunas* (NBC), *Número de Transições em Colunas* (NTC), *Número de Transições de Linhas* (NTL), *Somatório de Alturas* (SA) e *Intervalo de Alturas* (IA).

O segundo subgrupo faz uso somente das métricas propostas nesse trabalho, não avaliadas anteriormente, com objetivo de verificar se essas permitem que o agente extraia informações suficientes para obter resultados relevantes sem o uso de métricas já estabelecidas. As métricas que compõem esse subgrupo são aquelas de número, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17. As quais são, *Centro de Massa Vertical* (CMV), *Centro de Massa Horizontal* (CMH), *Porcentagem de Preenchimento* (PP), *Entropia do Tabuleiro* (ET), *Chance de Trava em Cantos* (CTC), *Área Máxima Contínua* (AMAC), *Área Mínima Contínua* (AMIC), *Potencial de Projeção* (PDP), *Simetria do Tabuleiro* (ST), *Número de Possíveis Posições* (NPP).

O terceiro subgrupo é composto pela união dos subconjuntos do primeiro e do segundo grupo, todas as métricas apresentadas (números 1 a 17), para disponibilizar o

maior conjunto de informação possível para otimização por parte do agente. Todas as métricas numeradas estão presentes.

O quarto subgrupo é composto pela análise do terceiro subgrupo e uso das 7 métricas com valores absolutos para seus pesos com maior valor absoluto após a otimização para o grupo 3 de dada avaliação. Dessa forma, cada uma das 3 diferentes avaliações, V_m , V_r e V_s , possui métricas diferentes para esse subgrupo. Esse grupo é desenvolvido para obter um desempenho mais otimizado através da remoção de métricas potencialmente ineficazes, para a otimização ocorrer sobre um espaço de busca com menor dimensionalidade.

As métricas presentes são, para V_m , 2, 4, 5, 10, 11, 15, 16. Significando, *Número Total de células poço (NTCP)*, *Número de Transições em Colunas (NTC)*, *Número de Transições em linhas (NTL)*, *Porcentagem de Preenchimento (PP)*, *Entropia do Tabuleiro (ET)*, *potencial de projeção (PDP)* e *Simetria do Tabuleiro (ST)*.

Para V_r as métricas são, 1, 2, 3, 7, 8, 9, 10. Ou seja, *Altura Total de Fixagem (ATF)*, *Número Total de Células Poço (NTCP)*, *Número de Buracos em Colunas (NBC)*, *o intervalo de alturas (IA)*, *Centro de Massa Vertical (CMV)*, *Centro de Massa Horizontal (CMH)* e *aPorcentagem de Preenchimento (PP)*.

Por fim, para V_s , as métricas utilizadas são, 1, 4, 5, 8, 9, 13, 17. Sendo essas, *Altura Total de Fixagem (ATF)*, *Número de Transições em Colunas (NTC)*, *Número de Transições de Linhas (NTL)*, *Centro de Massa Vertical (CMV)*, *Centro de Massa Horizontal (CMH)* *Área Máxima Contínua (AMAC)* e *Número de Possíveis Posições (NPP)*.

O quinto subgrupo é definido em mesma linha de pensamento, a conter as 4 métricas mais importantes identificadas na otimização do terceiro grupo, para que se teste os limites em que é possível obter um bom desempenho, com o mínimo de métricas definidas no espaço de busca.

5 RESULTADOS E DISCUSSÃO

Nesta seção, são discutidos os resultados obtidos durante toda a produção do trabalho, apresentando os melhores subconjuntos com seus relativos pesos e organizando as relevâncias encontradas, além disso, descrições visuais em forma de gráficos e tabelas são desenvolvidas por parte do autor para facilitar a compreensão dos resultados.

5.1 AVALIAÇÃO DOS RESULTADOS

Após a realização dos experimentos e os resultados obtidos após a efetuação dos experimentos são mostrados, para cada um deles, os diferentes grupos são apresentados e algumas estatísticas são apresentadas.

Nessa seção são apresentados os resultados para os diferentes grupos, apresentando as possíveis combinações obtidas após o treinamento. Além disso, são apresentados para as avaliações, V_m , V_r , e V_s os pesos atribuídos para o subgrupo que tenha atingido o desempenho mais satisfatório, os pesos otimizados para os outros subgrupos para dada avaliação estão disponíveis para o leitor no Apêndice A.

5.1.1 Avaliação dos subgrupos para a avaliação V_m

A Tabela 2 apresenta os resultados de diferentes subgrupos para a avaliação pela média.

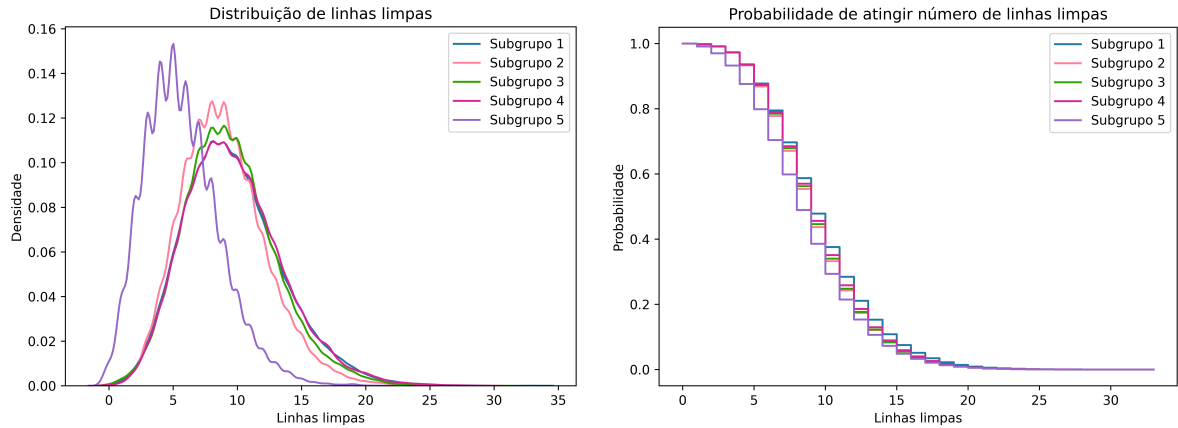
Tabela 2 – Desempenho dos subgrupos para avaliação por média

Estatística	Subgrupo 1	Subgrupo 2	Subgrupo 3	Subgrupo 4	Subgrupo 5
Min	0	0	0	0	0
Mean	9,68	8,94	9,47	9,73	5,71
Max	33	27	30	29	21
STD	3,82	3,30	3,54	3,76	2,92

Fonte: Elaborado pelo autor (2024).

Ainda, a Figura 12, apresenta os resultados atingidos nos testes para os diferentes subgrupos avaliados, no qual o subgrupo 5 tem distribuição de médias reduzido em relação aos outros subgrupos. Ainda o desempenho dos outros grupos é semelhante, sendo que aquele com melhor desempenho o subgrupo número 4 e aquele de segundo melhor desempenho, o de número 3.

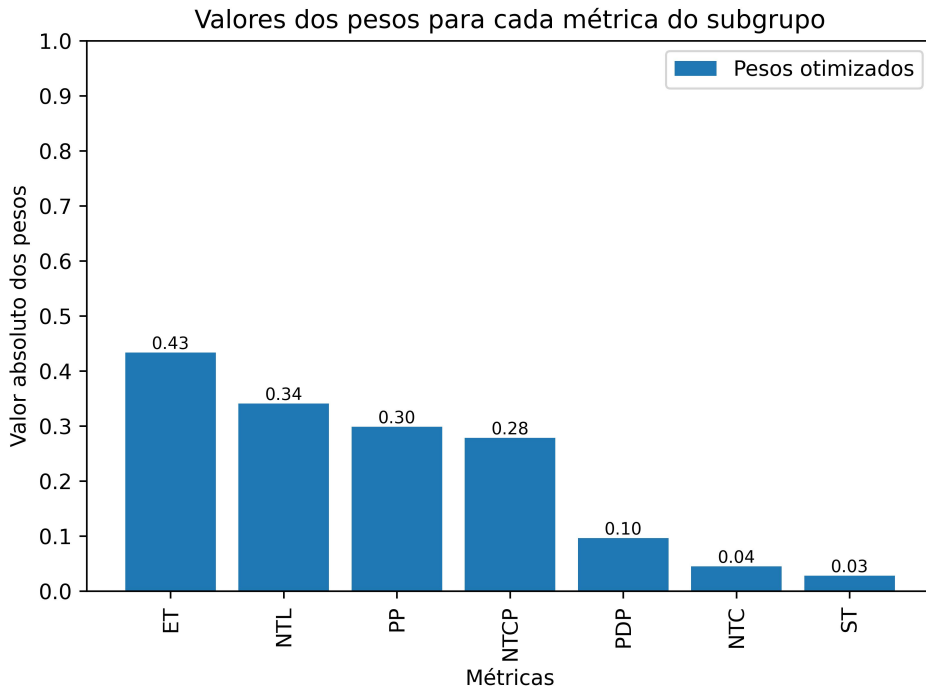
Figura 12 – Distribuição de linhas limpas para a avaliação média



Fonte: Elaborado pelo autor (2024).

Ademais, a Figura 13 mostra as métricas otimizadas para o subgrupo 4 da avaliação média, em que a métrica mais importante é a entropia do tabuleiro, seguida do número de transições em linha e a porcentagem de preenchimento. As métricas menos relevantes segundo o algoritmo de otimização tratam-se do potencial de projeção, número de transições por coluna e simetria do tabuleiro.

Figura 13 – Otimização das métricas para o Subgrupo 4 para a avaliação média



Fonte: Elaborado pelo autor (2024).

5.1.2 Avaliação dos subgrupos para a avaliação V_r

Na Tabela 3 são apresentados os resultados atingidos após o treinamento dos diferentes subgrupos definidos.

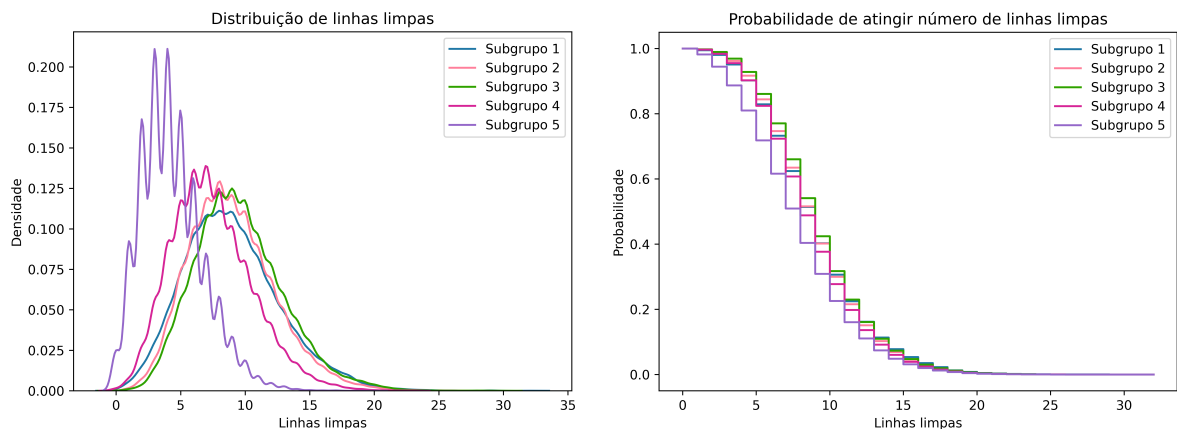
Tabela 3 – Desempenho dos subgrupos para avaliação por recorde

Estatística	Subgrupo 1	Subgrupo 2	Subgrupo 3	Subgrupo 4	Subgrupo 5
Min	0	0	0	0	0
Mean	8,95	8,96	9,54	7,41	4,35
Max	32	27	30	23	20
STD	3,72	3,32	3,41	3,11	2,40

Fonte: Elaborado pelo autor

Além da tabela provida, é possível avaliar os resultados dos diferentes subgrupos para a avaliação de recorde, através da Figura 14.

Figura 14 – Distribuição de linhas limpas para a avaliação recorde

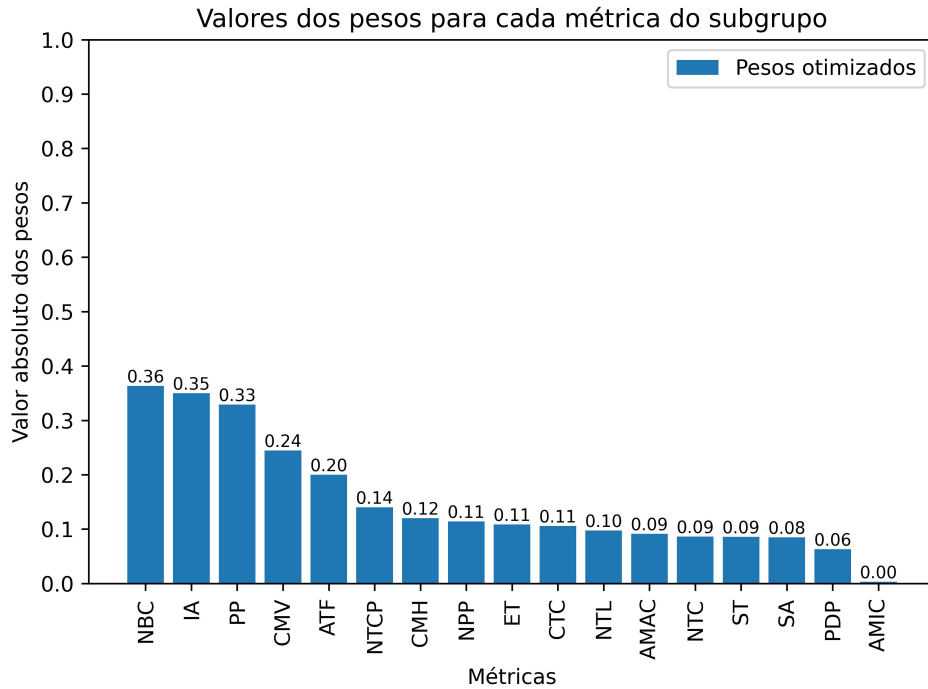


Fonte: Elaborado pelo autor (2024).

Por meio da Figura 14, é possível inferir um desempenho semelhante dentre os subgrupos avaliados, sendo o subgrupo 5 aquele de pior desempenho em média, enquanto o subgrupo 4 tem desempenho intermediário entre os melhores grupos e pior grupo. Os subgrupos 2 e 3 aqueles de melhor desempenho, o que pode significar assim, que as métricas do subgrupo 1 e 2 são informativas, e a redução dimensional do espaço é um método ineficiente para garantir um melhor desempenho em menor complexidade computacional.

Ainda, a Figura 15 mostra as métricas otimizadas com o eixo das ordenadas com o valor absoluto do peso otimizado, de 0 a 1, enquanto o eixo das abcissas é ordenado, da esquerda a direita em ordem decrescente para os pesos, para o subgrupo 3.

Figura 15 – Otimização das métricas para o Subgrupo 3 para a avaliação recorde



Fonte: Elaborado pelo autor (2024).

Para a Figura 15, é possível obter que as duas métricas mais importantes são aquelas de, número de buracos em colunas, intervalo de alturas e porcentagem de preenchimento, com mais de 2 ordens de importância da sétima métrica. As três métricas menos importantes foram otimizadas a ser, o somatório de alturas, potencial de projeção e área mínima contínua.

5.1.3 Avaliação dos subgrupos para a avaliação V_s

Por fim, a Tabela 4 apresenta os resultados para os resultados dos diferentes subgrupos para a avaliação ponderada, a distribuição da avaliação ponderada é vista também na Figura 16.

Tabela 4 – Desempenho dos subgrupos para avaliação ponderada

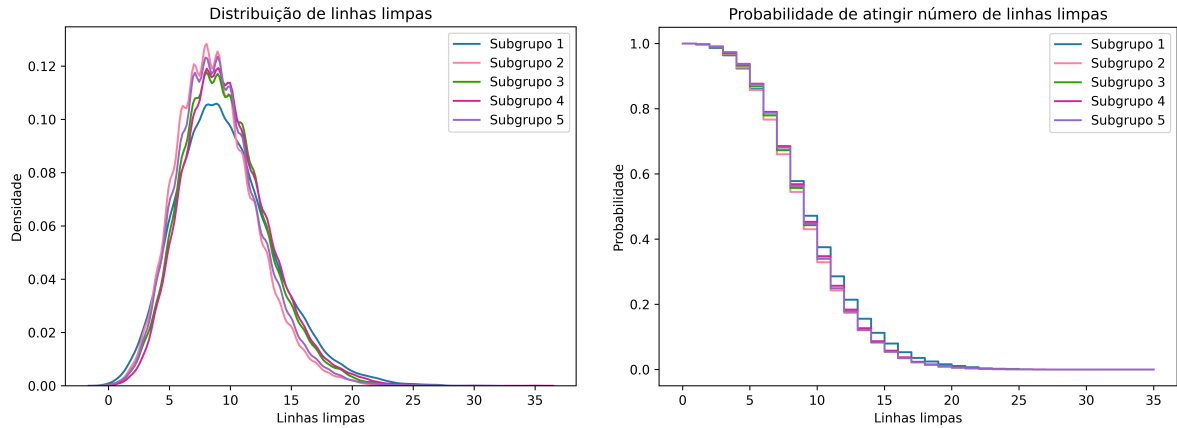
Estatística	Subgrupo 1	Subgrupo 2	Subgrupo 3	Subgrupo 4	Subgrupo 5
Min	0	0	1	0	0
Média	9,62	8,87	9,52	9,72	9,12
Max	31	27	28	35	31
STD	3,94	3,28	3,50	3,53	3,34

Fonte: Elaborado pelo autor (2024).

Para os subgrupos avaliados com uma pontuação ponderada entre recorde e média, aquele que obteve o pior desempenho foi o segundo subgrupo, enquanto aquele

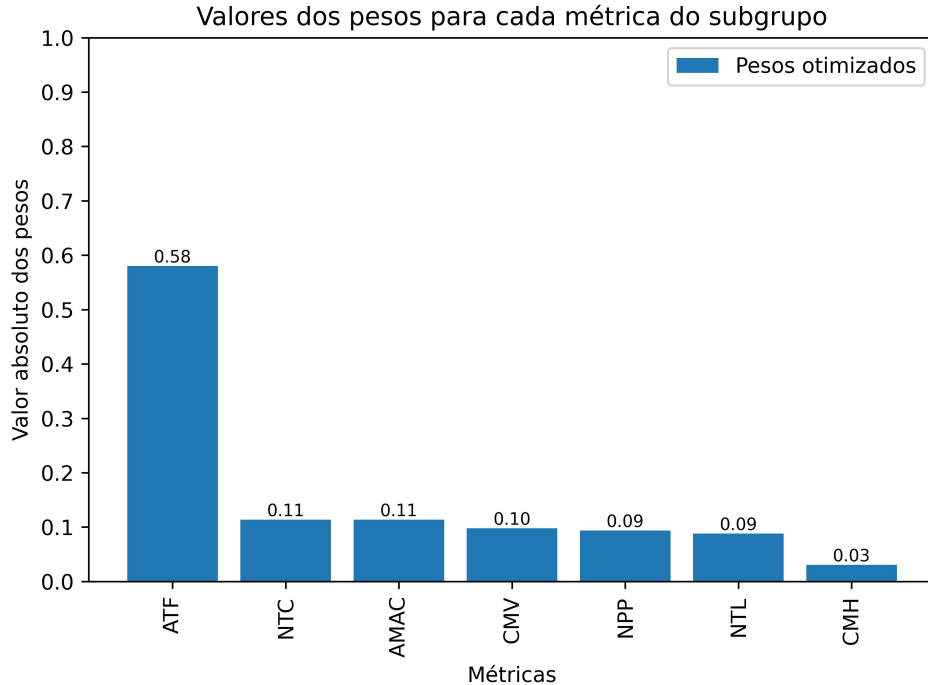
com melhor desempenho foi o subgrupo 4, que contém um subgrupo de métricas do subgrupo 3.

Figura 16 – Distribuição de linhas limpas para a avaliação ponderada



Fonte: Elaborado pelo autor (2024).

Figura 17 – Otimização das métricas para o Subgrupo 4 para a avaliação ponderada



Fonte: Elaborado pelo autor (2024).

Após o processo de otimização, a mais importante das métricas é aquela dada pela altura de fixagem das peças, o número de transições em coluna e área máxima contínua. As três métricas menos relevantes são o número de posição de peças, o número de transições em linhas e o centro de massa horizontal.

No total, foram avaliados 5 subgrupos, em 3 diferentes condições de avaliação. Em que, ao comparar as métricas mais importantes, percebe-se o padrão de que para os subgrupos mais performáticos de cada avaliação, certas métricas são dadas como informativas pelo algoritmo de otimização, aparecendo em mais de uma avaliação, no grupo mais performático, como uma das cinco métricas de maior peso absoluto. Essas são, a altura total de fixagem (ATF), centro de massa vertical (CMV) e a porcentagem de preenchimento (PP).

Por fim, as Tabelas 5, 6 e 7, se apresentam uma comparação para os subgrupos para os diferentes tipos de avaliação ao qual foram submetidos, os grupos selecionados para essa comparação são o subgrupo 3, por utilizar todas as métricas, e os subgrupos 4 e 5 por terem as maiores diferenças entre as diferentes avaliações. As tabelas para todos os subgrupos podem ser encontradas no Apêndice B.

Tabela 5 – Desempenho das diferentes avaliações do subgrupo 3

Estatística	V_m	V_r	V_s
Min	0	0	1
Média	9,47	9,54	9,52
Max	30	30	28
STD	3,54	3,41	3,50

Fonte: Elaborado pelo autor (2024).

A Tabela 5 indica que o conjunto de métricas utilizado pelo subgrupo 3 possui poucas diferenças entre os diferentes tipos de avaliação ao qual foi submetido, sendo a melhor avaliação para esse a avaliação recorde.

Tabela 6 – Desempenho das diferentes avaliações do subgrupo 4

Estatística	V_m	V_r	V_s
Min	0	0	0
Média	9,73	7,41	9,72
Max	29	23	35
STD	3,76	3,11	3,53

Fonte: Elaborado pelo autor (2024).

A Tabela 6 indica que o conjunto de métricas utilizado pelo subgrupo 4 é afetado de forma mais acentuada que o subgrupo 3. A pior avaliação para esse subgrupo é V_r , potencialmente por conta do menor número de métricas, que faz com que situações específicas impliquem em decisões sub-ótimas durante a tentativa de maximizar o número de linhas removidas.

O subgrupo 5 apresenta a maior discrepância entre os diferentes subgrupos avaliados, o que pode indicar que a redução de métricas, pode ser eficiente para um

Tabela 7 – Desempenho das diferentes avaliações do subgrupo 5

Estatística	V_m	V_r	V_s
Min	0	0	0
Média	5,71	4,35	9,12
Max	21	20	31
STD	2,92	2,40	3,34

Fonte: Elaborado pelo autor (2024).

espaço de busca mais eficiente, no entanto, arrisca-se um desempenho instável, assim como ocorreu para o subgrupo 4, na avaliação de recorde.

Comparando os diferentes subgrupos apresentados, infere-se que o subgrupo 3 possui comportamento semelhante para os diferentes tipos de avaliações para o qual foi submetido, enquanto o subgrupo 4 e 5, enquanto capazes de atingir bom desempenho, são mais sujeitos a instabilidade, por conta do reduzido número de métricas.

6 CONCLUSÕES

O trabalho visou, avaliou, propôs e testou novas métricas, a avaliar seu potencial de uso para a tomada de decisão para possíveis problemas semelhantes. Os resultados atingidos são satisfatórios, por demonstrarem a viabilidade do uso de diferentes estratégias para a determinação de um agente para a tarefa de limpeza de linhas no tabuleiro, um problema que pode ser também visto semelhantemente a problemas industriais (SOSTER; BIRKEN; JASKOWIAK, 2021).

Com o desenvolvimento dos diferentes grupos, infere-se que os diferentes métodos de avaliação do desempenho do agente não afetam significativamente o agente final, mantendo um desempenho semelhante para um mesmo entre as diferentes formas de avaliação testadas. Ainda, a redução do número de parâmetros é uma forma minimamente eficaz para a melhora da qualidade da solução, sendo majoritariamente útil para reduzir a complexidade da solução encontrada para futuros trabalhos e entendimento geral do problema, que para melhora da qualidade do modelo em si.

Logo definem-se satisfeitos os objetivos de estudo de métricas para um ensaio em métricas estudadas especificadamente no contexto sem rotações uma vez que, com esse há boas indicações de heurísticas que podem ser acompanhadas para a geração de outros conjuntos por meio da investigação dos resultados atingidos e apresentados nos capítulos anteriores.

Uma das limitações encontrada nesse trabalho trata-se das limitações de hardware, que não permitem um aumento extremamente significativo do número de partículas, em razão das demoradas execuções necessárias para avaliar cada subconjunto de métricas testado, mesmo que um número maior de partículas poderia implicar no escape de pontos críticos locais por conta da natureza estocástica do processo, de forma que esse pode obter melhores resultados. Ainda, a escassez de literatura para agentes de Tetris submetidos no contexto sem rotações, torna mais árdua a navegação por esse campo, um tópico que esse trabalho aspira abordar nas possibilidades do autor.

Como trabalhos futuros, uma possível abordagem do problema seria avaliar diferentes algoritmos de otimização, como redes neurais e algoritmos genéticos para alguns dos subconjuntos de desempenho mais significativo, juntamente com as diferentes formas de avaliação para verificar a influência de cada um desses itens sobre o agente. Outra possível influência, é a multivariabilidade introduzida pela avaliação ponderada, que poderia ser otimizada para o agente. Ainda um número de partículas dinâmico poderia ser usado para garantir uma busca eficiente pela quantidade diferenciada de métricas, que aumenta ou diminui a complexidade do espaço de busca.

Logo, apesar das limitações, a utilização de métricas se mostra um recurso desejado para problemas de otimização como este. Portanto, aspira-se que este trabalho possa ser usado para o desenvolvimento de trabalhos futuros.

REFERÊNCIAS

- ALGORTA, S.; ŞİMŞEK, Ö. The Game of Tetris in Machine Learning. 2019. Disponível em: <http://arxiv.org/abs/1905.01652>.
- BANGERT, P. Machine Learning. In: **Machine Learning and Data Science in the Oil and Gas Industry**. Elsevier, 2021. p. 37–67. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/B9780128207147000030>.
- BERECZKI, Z. **NES Tetris NoRo lvl 0 start 72 lines [World Record]**. 2023. Disponível em: <https://www.youtube.com/watch?v=l7uKLk97xNg>. Acesso em: 01 maio 2023.
- BIRKEN, M. **Applying Artificial Intelligence to Handicapping NES Tetris**. 2019. Disponível em: <https://meatfighter.com/handicappedtetris>. Acesso em: 04 jun. 2023.
- BÖHM, N.; KÓOKAI, G.; MANDL, S. An Evolutionary Approach to Tetris. **Proceedings of the 6th Metaheuristics International Conference (MIC2005)**, p. 137–148, 2005. ISSN 00222429.
- CTWC. **Classic Tetris World Championship**. 2023. Disponível em: <https://thectwc.com/>. Acesso em: 30 abr. 2023.
- DEMAINE, E. D.; HOHENBERGER, S.; LIBEN-NOWELL, D. Tetris is hard, even to approximate. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 2697, p. 351–363, 2003.
- FAHEY, C. **Tetris**. 2003. Disponível em: <https://www.colinfahey.com/tetris/tetris.html>. Acesso em: 04 jun. 2023.
- GAD, A. G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. **Archives of Computational Methods in Engineering**, Springer Netherlands, v. 29, n. 5, p. 2531–2561, 2022. ISSN 18861784. Disponível em: <https://doi.org/10.1007/s11831-021-09694-4>.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **Proceedings of ICNN'95 - INTERNATIONAL CONFERENCE ON NEURAL NETWORK**. IEEE, v. 4, p. 1942–1948, 1995. Disponível em: <http://ieeexplore.ieee.org/document/488968/>. Acesso em: 21 jun. 2023.
- NAREYEK, A. Review: Intelligent Agents for Computer Games. In: **Proceedings of INTERNATIONAL CONFERENCE ON COMPUTERS AND GAMES**. v. 2063, p. 414–422, 2001. Disponível em: http://link.springer.com/10.1007/3-540-45579-5_28.
- OLIVEIRA, J. F. et al. A survey on heuristics for the two-dimensional rectangular strip packing problem. **Pesquisa Operacional**, v. 36, n. 2, p. 197–226, aug. 2016.
- OPENAI et al. Dota 2 with Large Scale Deep Reinforcement Learning. 2019. Disponível em: <http://arxiv.org/abs/1912.06680>.
- POLI, R.; KENNEDY, J.; BLACKWELL, T. Particle swarm optimization: An overview. **Swarm Intelligence**, v. 1, n. 1, p. 33–57, 2007.

Python Software Foundation. **Welcome to Python.org**. 2024. Disponível em: <https://www.python.org/>.

ROCHOL, J. **Comunicação de Dados - Vol. 22: Série Livros Didáticos Informática UFRGS**. Bookman, 2009. (Livros Didáticos Informática UFRGS). ISBN 9788540700536. Disponível em: <https://books.google.com.br/books?id=UP4YD7eoaDEC>.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach, Global Edition**. Pearson Education, 2016. ISBN 9781292153971. Disponível em: https://books.google.com.br/books?id=_BV6DAAAQBAJ.

SOSTER, A. A. L.; BIRKEN, M.; JASKOWIAK, P. A. Playing NES Tetris with No piece rotations. *In: XX SIMPÓSIO BRASILEIRO DE GAMES E ENTRETENIMENTO DIGITAL. Anais*. p. 339–343, oct. 2021. Disponível em: https://sol.sbc.org.br/index.php/sbgames_estendido/article/view/19664. Acesso em: 20 jun. 2023.

Tetris Holding. **Tetris**. 2023. Disponível em: <https://tetris.com/>. Acesso em: 14 maio 2023.

The Apache Software Foundation. **Maven Project**. 2024. Disponível em: <https://maven.apache.org/>.

The Matplotlib development team. **Matplotlib: Visualization with Python**. 2024. Disponível em: <https://matplotlib.org/>.

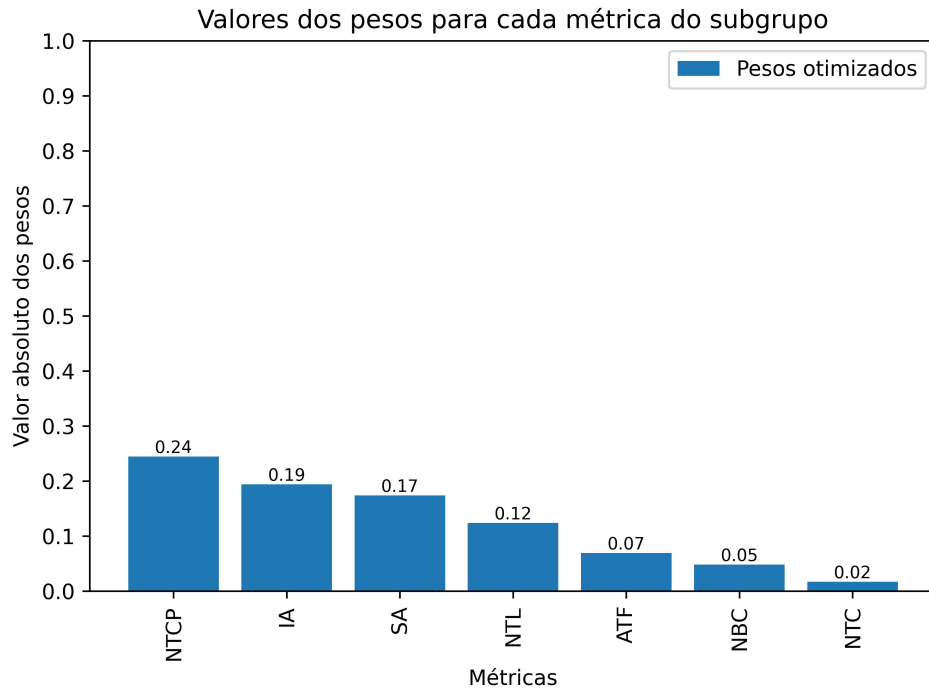
THIERY, C.; SCHERRER, B. Building Controllers for Tetris. **ICGA Journal**, v. 32, n. 1, p. 3–11, mar. 2009.

Valve Corporation. **Dota 2**. 2024. Disponível em: <https://www.dota2.com/home>.

WANG, D.; TAN, D.; LIU, L. Particle swarm optimization algorithm: an overview. **Soft Computing**, v. 22, n. 2, p. 387–408, 2018.

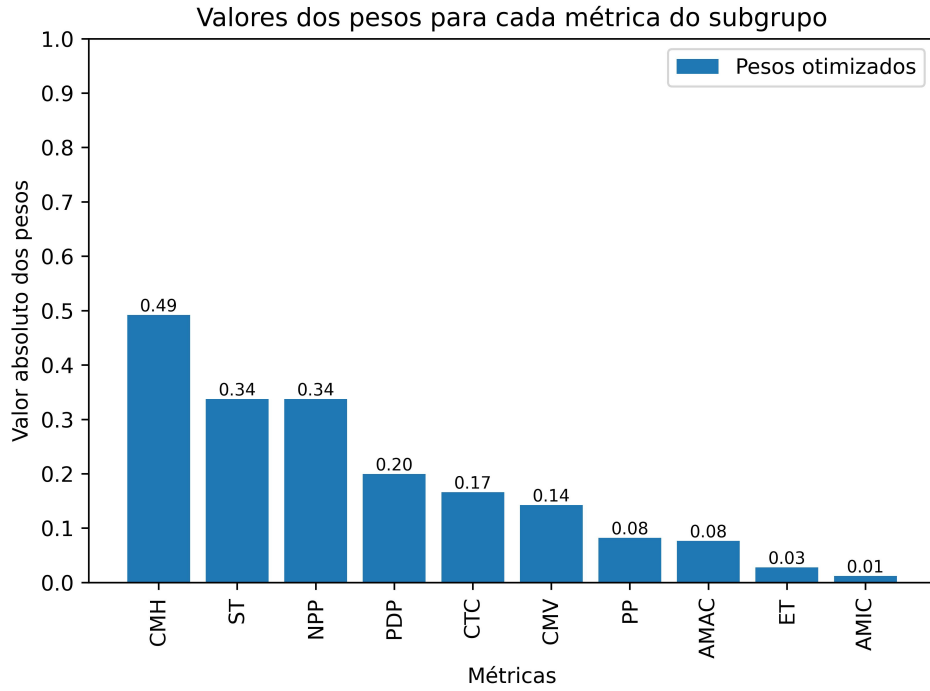
APÊNDICE A - PESOS RESULTANTES PARA CADA SUBGRUPO

Figura 18 – Otimização das métricas para o Subgrupo 1 para a avaliação média



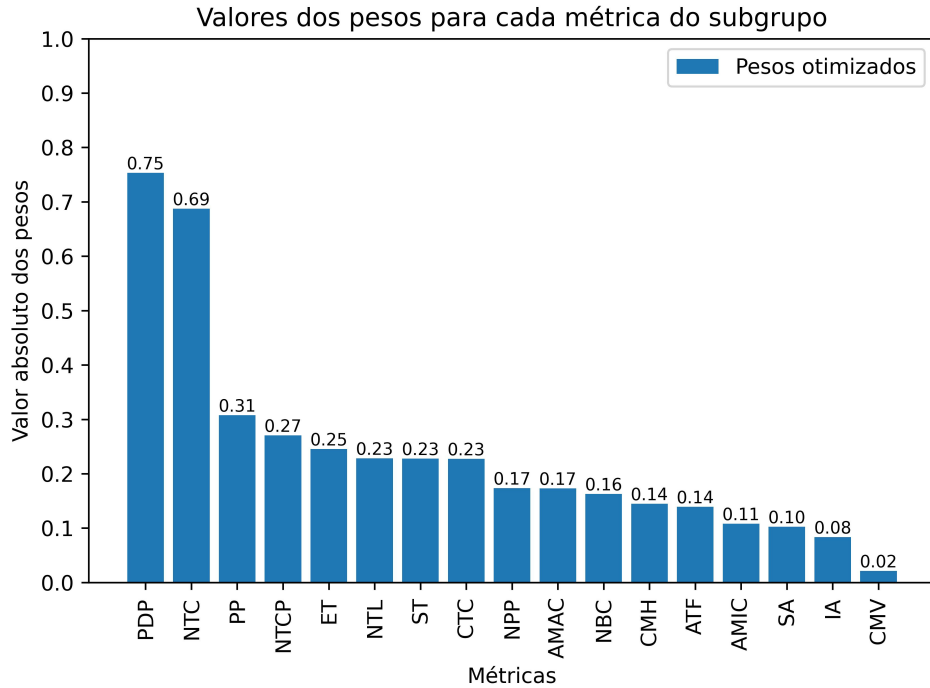
Fonte: Elaborado pelo autor (2024).

Figura 19 – Otimização das métricas para o Subgrupo 2 para a avaliação média



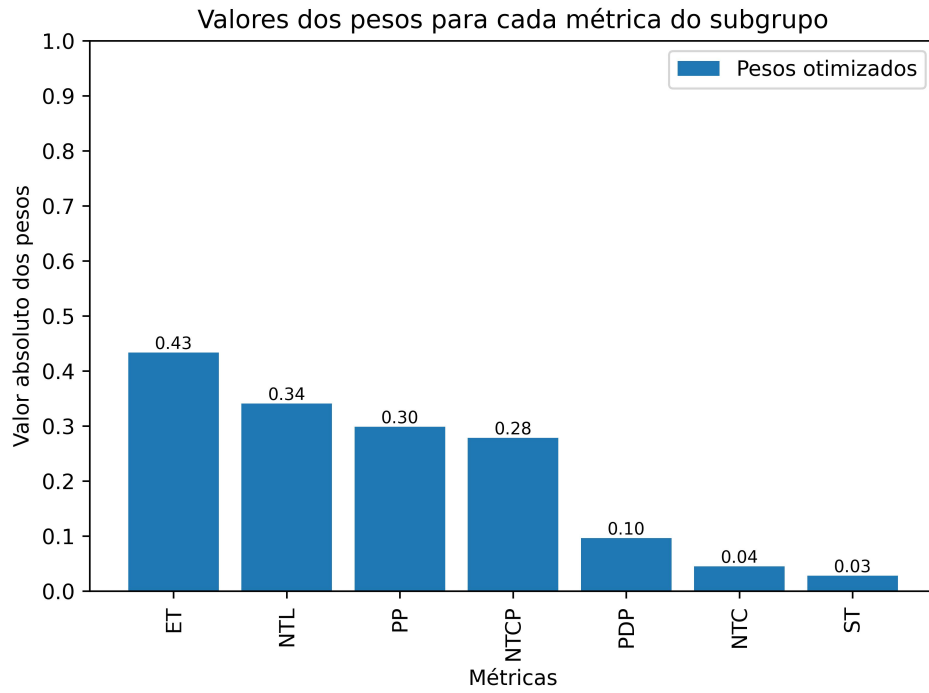
Fonte: Elaborado pelo autor (2024).

Figura 20 – Otimização das métricas para o Subgrupo 3 para a avaliação média



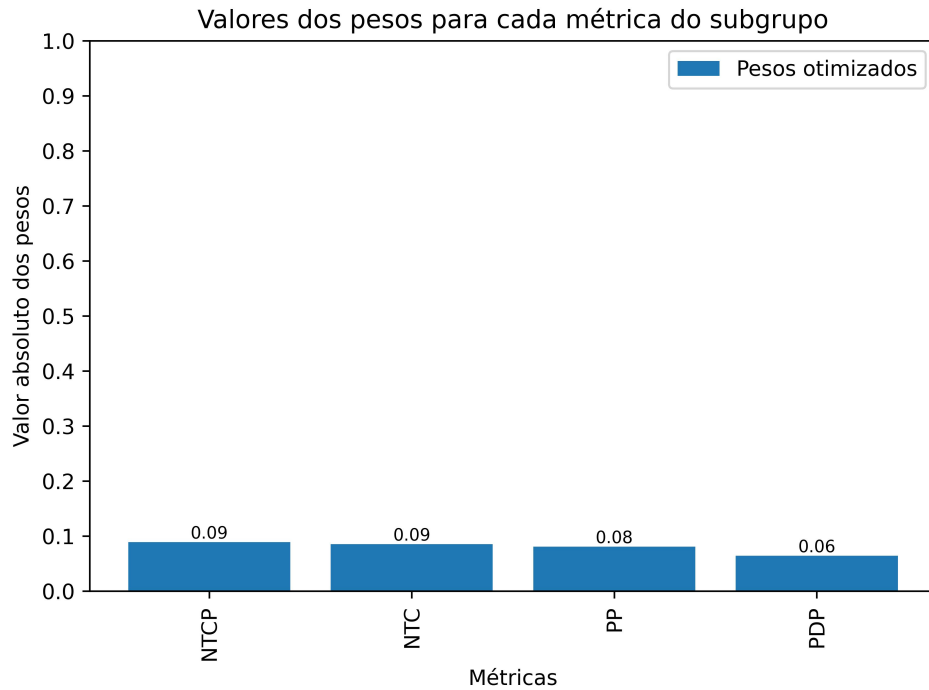
Fonte: Elaborado pelo autor (2024).

Figura 21 – Otimização das métricas para o Subgrupo 4 para a avaliação média



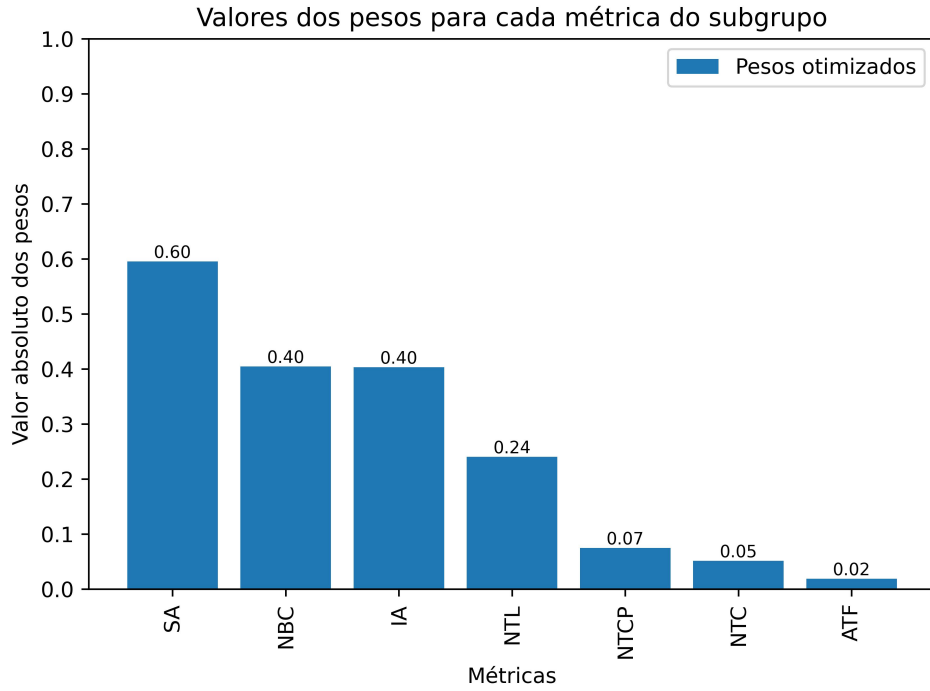
Fonte: Elaborado pelo autor (2024).

Figura 22 – Otimização das métricas para o Subgrupo 5 para a avaliação média



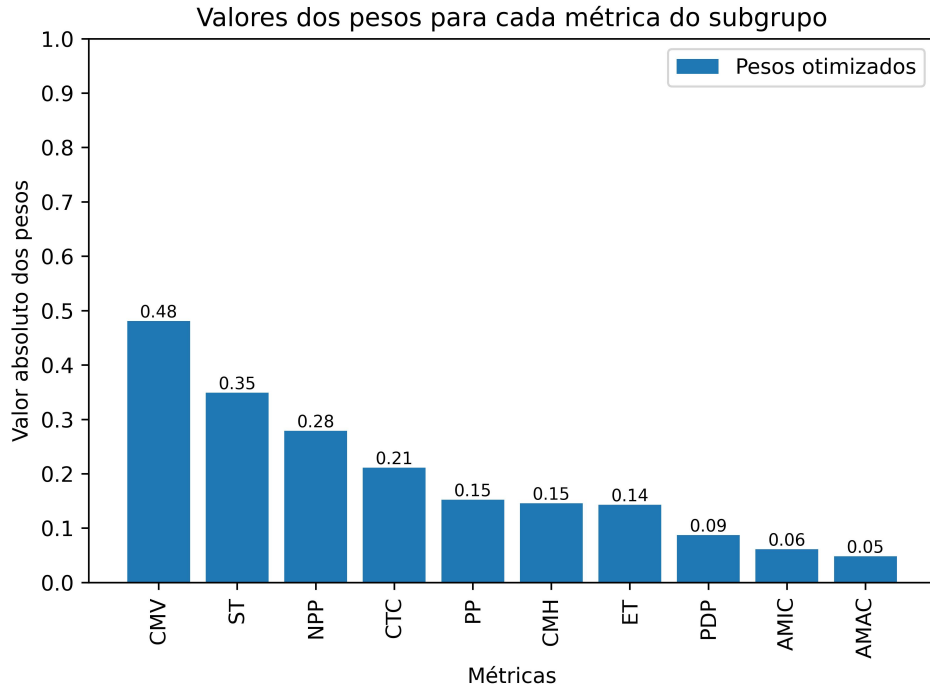
Fonte: Elaborado pelo autor (2024).

Figura 23 – Otimização das métricas para o Subgrupo 1 para a avaliação recorde



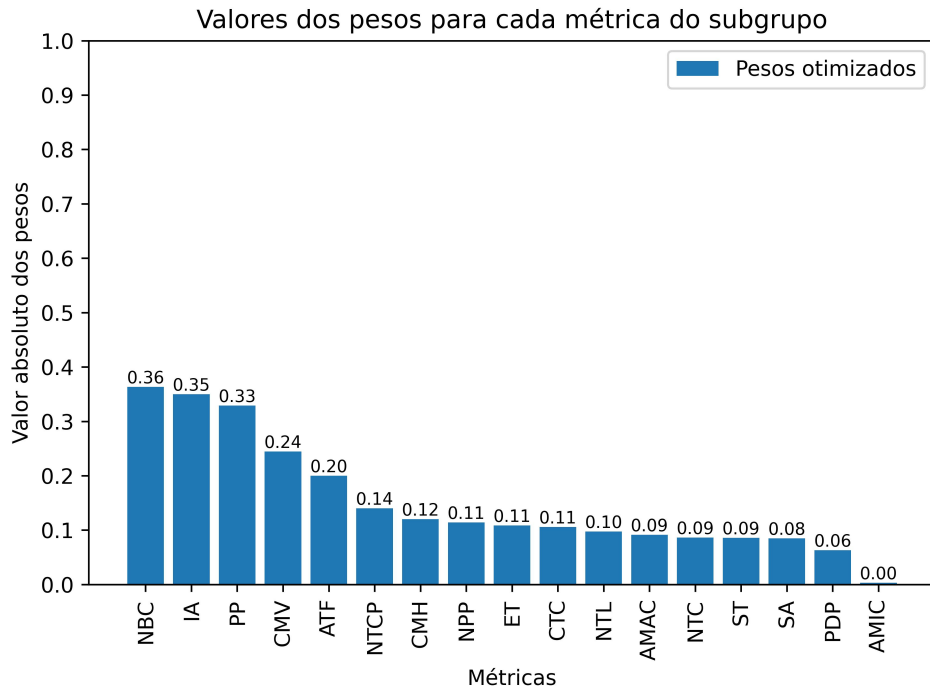
Fonte: Elaborado pelo autor (2024).

Figura 24 – Otimização das métricas para o Subgrupo 2 para a avaliação recorde



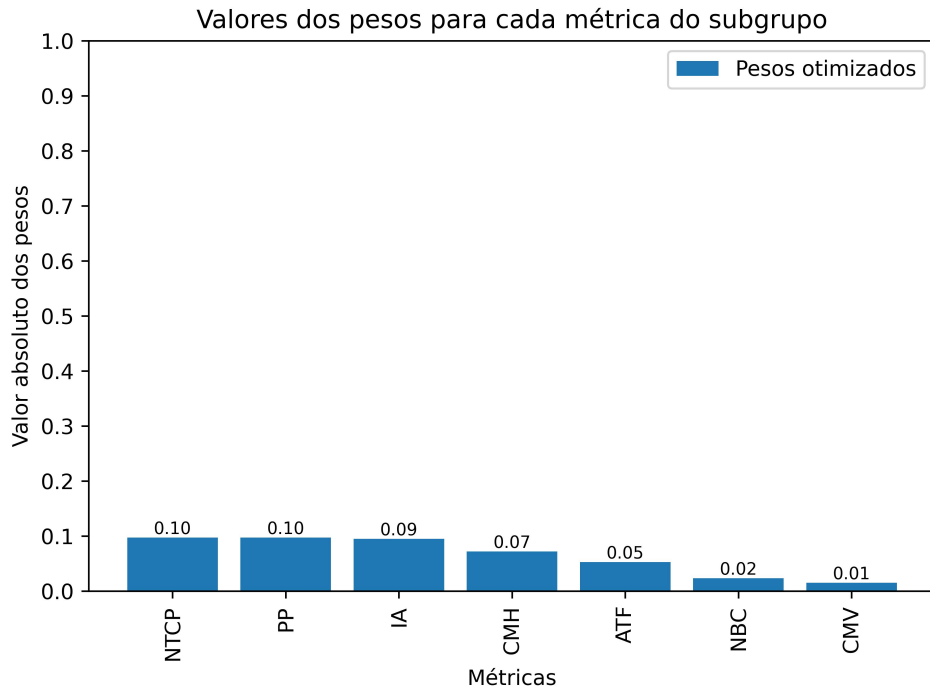
Fonte: Elaborado pelo autor (2024).

Figura 25 – Otimização das métricas para o Subgrupo 3 para a avaliação recorde



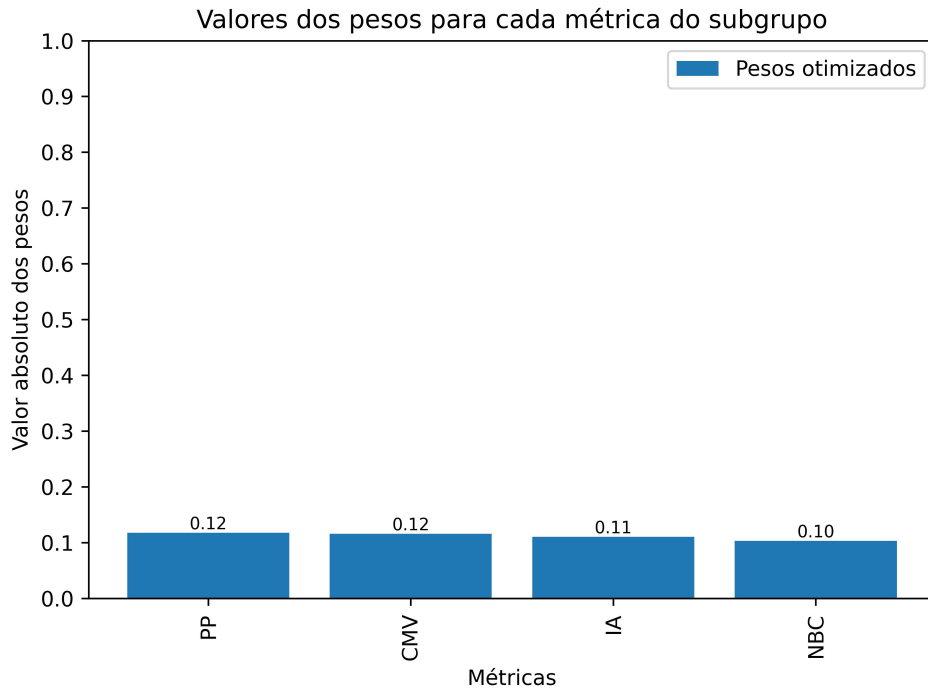
Fonte: Elaborado pelo autor (2024).

Figura 26 – Otimização das métricas para o Subgrupo 4 para a avaliação recorde



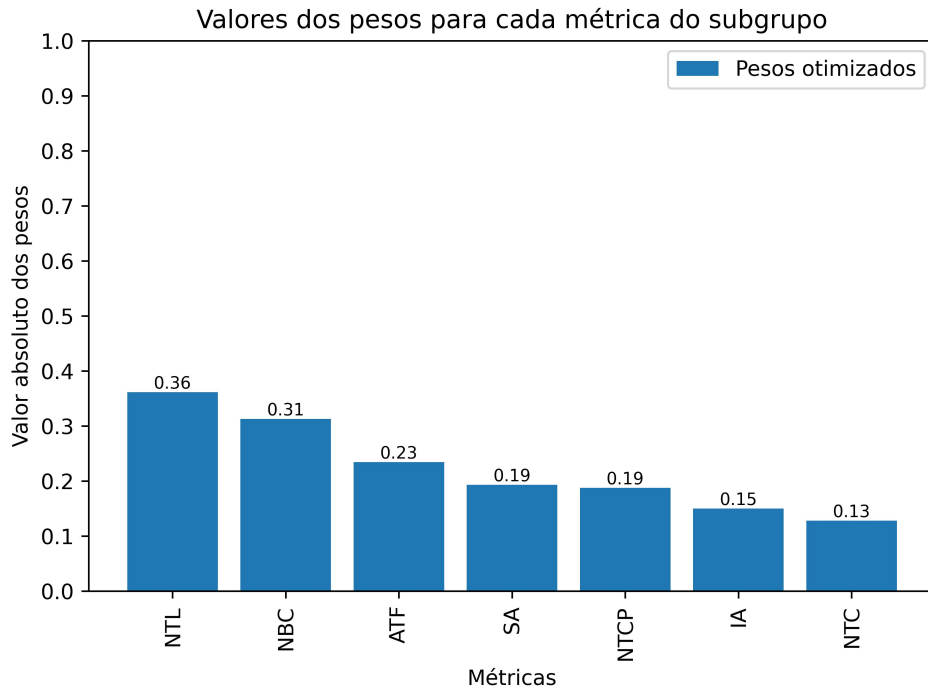
Fonte: Elaborado pelo autor (2024).

Figura 27 – Otimização das métricas para o Subgrupo 5 para a avaliação recorde



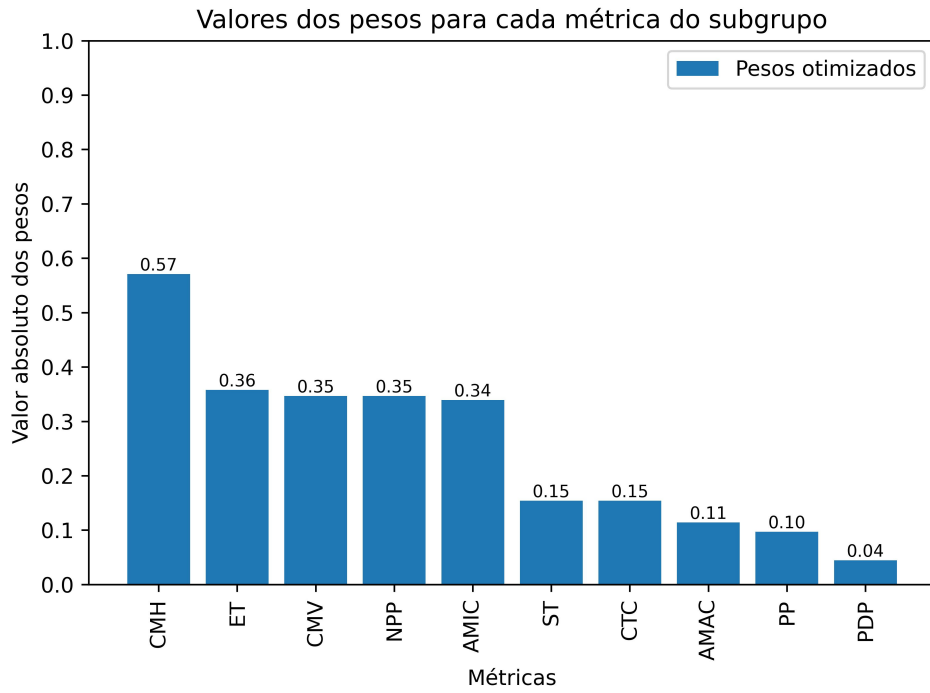
Fonte: Elaborado pelo autor (2024).

Figura 28 – Otimização das métricas para o Subgrupo 1 para a avaliação ponderada



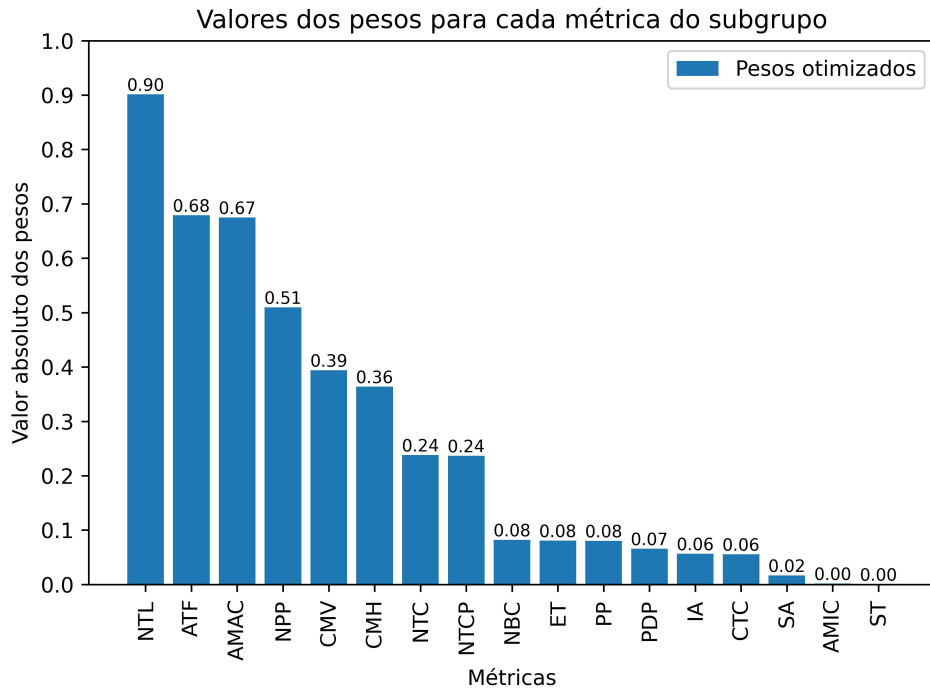
Fonte: Elaborado pelo autor (2024).

Figura 29 – Otimização das métricas para o Subgrupo 2 para a avaliação ponderada



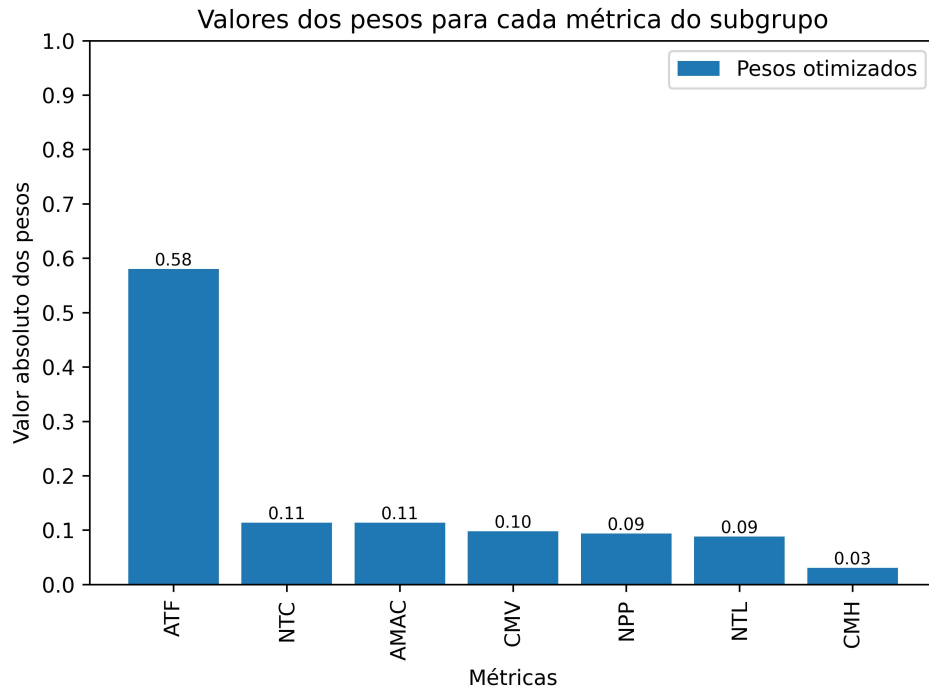
Fonte: Elaborado pelo autor (2024).

Figura 30 – Otimização das métricas para o Subgrupo 3 para a avaliação ponderada



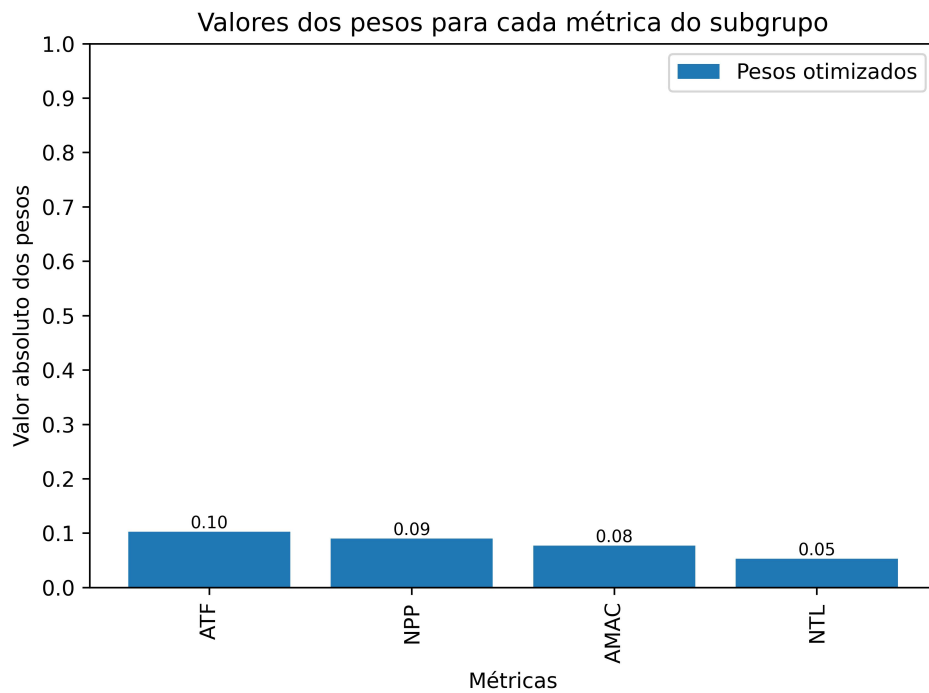
Fonte: Elaborado pelo autor (2024).

Figura 31 – Otimização das métricas para o Subgrupo 4 para a avaliação ponderada



Fonte: Elaborado pelo autor (2024).

Figura 32 – Otimização das métricas para o Subgrupo 5 para a avaliação ponderada



Fonte: Elaborado pelo autor (2024).

APÊNDICE B - COMPARAÇÕES ENTRE AVALIAÇÕES

Tabela 8 – Desempenho das diferentes avaliações do subgrupo 1

Estatística	V_m	V_r	V_s
Min	0	0	0
Média	9,68	8,95	9,62
Max	33	32	31
STD	3,82	3,72	3,94

Fonte: Elaborado pelo autor (2024).

Tabela 9 – Desempenho das diferentes avaliações do subgrupo 2

Estatística	V_m	V_r	V_s
Min	0	0	0
Média	8,94	8,96	8,87
Max	27	27	27
STD	3,30	3,32	3,28

Fonte: Elaborado pelo autor (2024).

Tabela 10 – Desempenho das diferentes avaliações do subgrupo 3

Estatística	V_m	V_r	V_s
Min	0	0	1
Média	9,47	9,54	9,52
Max	30	30	28
STD	3,54	3,41	3,50

Fonte: Elaborado pelo autor (2024).

Tabela 11 – Desempenho das diferentes avaliações do subgrupo 4

Estatística	V_m	V_r	V_s
Min	0	0	0
Média	9,73	7,41	9,72
Max	29	23	35
STD	3,76	3,11	3,53

Fonte: Elaborado pelo autor (2024).

Tabela 12 – Desempenho das diferentes avaliações do subgrupo 5

Estatística	V_m	V_r	V_s
Min	0	0	0
Média	5,71	4,35	9,12
Max	21	20	31
STD	2,92	2,40	3,34

Fonte: Elaborado pelo autor (2024).